Decompiling the Synergy: An Empirical Study of Human–LLM Teaming in Software Reverse Engineering

Zion Leonahenahe Basque*, Samuele Doria†, Ananta Soneji*, Wil Gibbs*, Adam Doupé*,
Yan Shoshitaishvili*, Eleonora Losiouk†, Ruoyu Wang*, Simone Aonzo‡

*Arizona State University
†University of Padua
‡EURECOM

{*zbasque, asoneji, wfgibbs, doupe, yans, fishw*}*@asu.edu*
sdoria@math.unipd.it, eleonora.losiouk@unipd.it
simone.aonzo@eurecom.fr

Software Reverse Engineering (SRE)

- Understanding Software → Securing Software

Software Reverse Engineering (SRE)

- U

Complex + Primarily a “human-driven process”

LLMs for Reverse Engineering?

LLMs for Reverse Engineering?

- Function Summarization (Description / Comment)
- Function Identification (Identifying well known algorithms)
- Function and Variable Renaming (FUNC_5888 -> connect)
- Vulnerability Identification
- …

What we do not consider

- Interaction between SRE practitioners and LLMs
  - Existing Studies Focus on LLM improvements to SRE in “isolation”
  - Existing Studies Fail to consider human-LLM dynamics involving iterative sub-tasks

So..

- This paper is the "**first work** studying the dynamics between humans and LLMs during SRE"

Overview

- Human Study
  - Measuring the performance of SRE practitioners with/without LLM assistance

Overview

- ance of SRE p M

What kind of LLM Assistance?

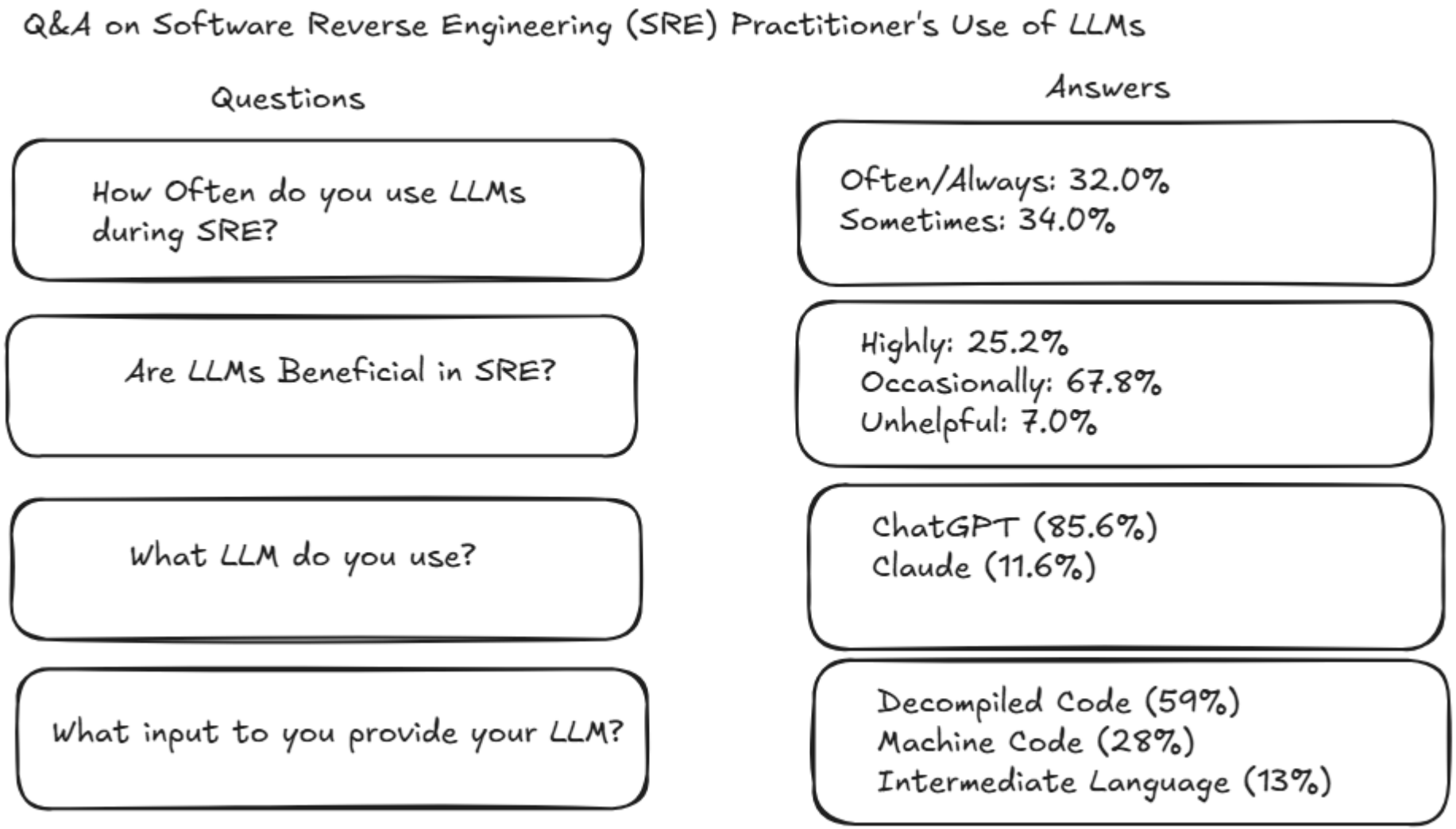
How do you “measure” performance in SRE?

What Kind of LLM Assistance?

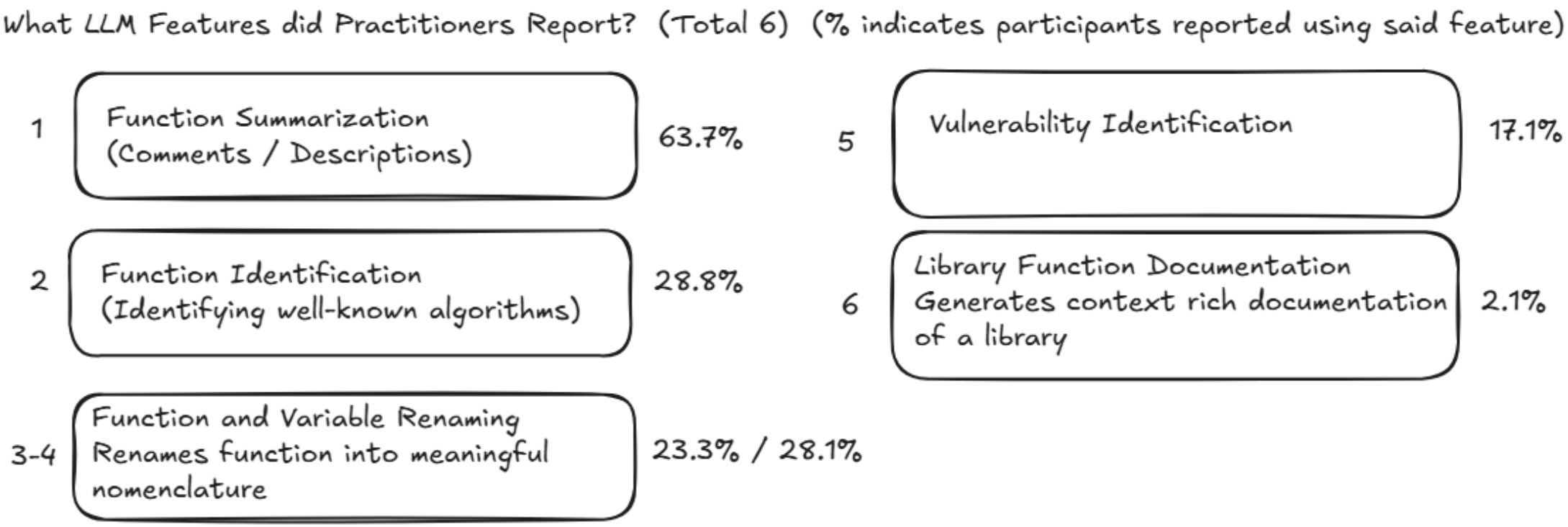
- Conduct ***Formative Research*** **(RQ1)**:
- "How do SRE practitioners integrate LLMs into SRE process, and what are their perceptions?"

- 153 Participants
  - **LLM Use?** How practitioners use LLMs
  - **LLM Features?** Kind of features do practitioners use
  - **LLM Perceptions?** What do practitioners think of LLMs

LLM Use


Q&A on Software Reverse Engineering (SRE) Practitioner's Use of LLMs
Questions
Answers
How Often do you use LLMs during SRE?
Often/Always: 32.0%
Sometimes: 34.0%
Are LLMs Beneficial in SRE?
Highly: 25.2%
Occasionally: 67.8%
Unhelpful: 7.0%
What LLM do you use?
ChatGPT (85.6%)
Claude (11.6%)
What input to you provide your LLM?
Decompiled Code (59%)
Machine Code (28%)
Intermediate Language (13%)

LLM Features



LLM Perception

Perception on LLMs (Free Responce)

Pros

-Excels at explaining behavior of decompiled Code

-Improves code readability (variable renaming)

-Useful for understanding known algorithms

- Accelerates workflow (generates scripts for SRE framework)

Cons

- Often produce incorrect of misleading responces
(reduce trust, waste time)

- Generic of Superficial explanation

- Effectiveness diminishes for large, obfuscated, or highly mathematical tasks

Overview

- ance of SRE p M

What kind of LLM Assistance?

Six LLM Features identified.

How do you “measure” performance in SRE?

Measuring Performance on SRE

- CTF Challenge (2 Questions)

- 48 Participants (subset of 153 in previous study)
  - 24 self reported novices
  - 24 self reported experts

Measu

- CTF Cha
- 48 Partic
  - 24 selt
  - 24 selt

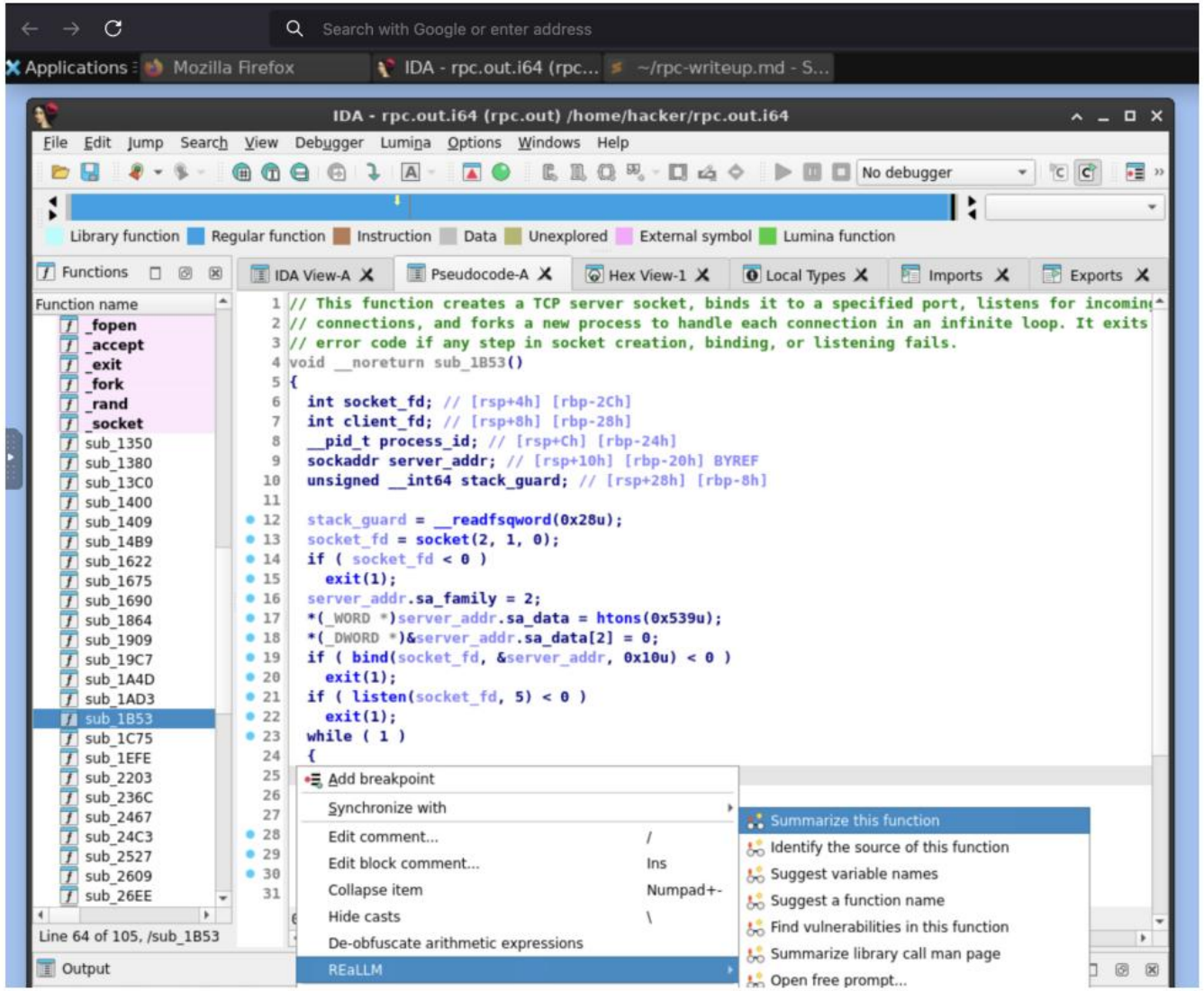
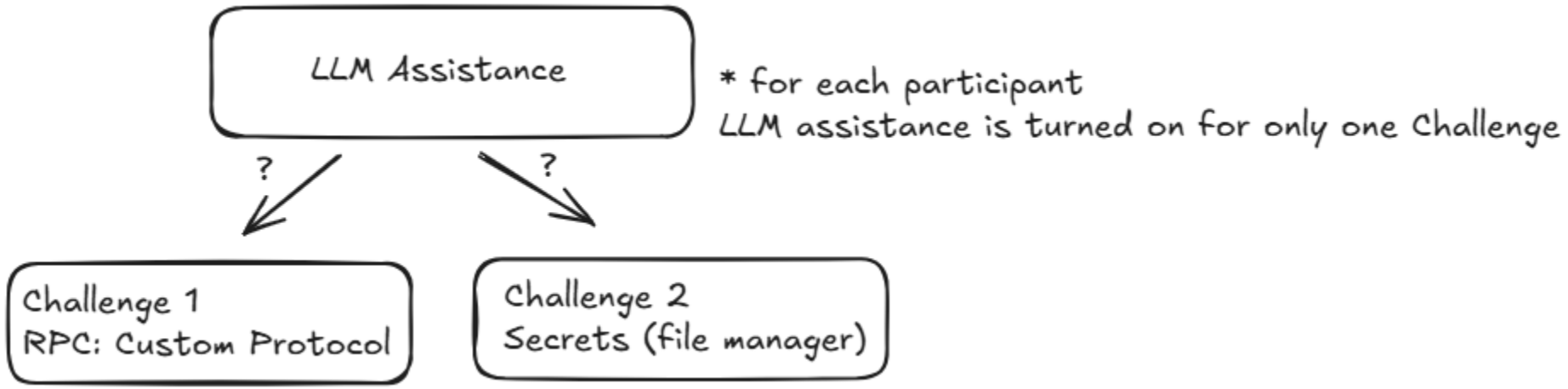


Figure 1: A screenshot of the online platform after a participant starts a challenge. A VNC is accessible in the browser that provides access to an instrumented decompiler (with an LLM plugin), browser, and text editor.

How is LLM Assistance applied?



How do you Measure?

Task (TODO)

- two CTF challenges (online)
- post-study survey
- two writeups describing their solutions

Assessment

Evaluation (0-8)

- Each challenge has 7 checkpoints
- 8 point for fully correct solution

e.g., Checkpoints (Challenge 1)

1,2: identify numeriacal operation
+ hard coded name

3. identify bug in RPC
4. mentioning "encrypted" or "hashed"

5,6,7: identifying three known algorithms
(Base64, RLE, TEA)

Overview

- ance of SRE p M

What kind of LLM Assistance?

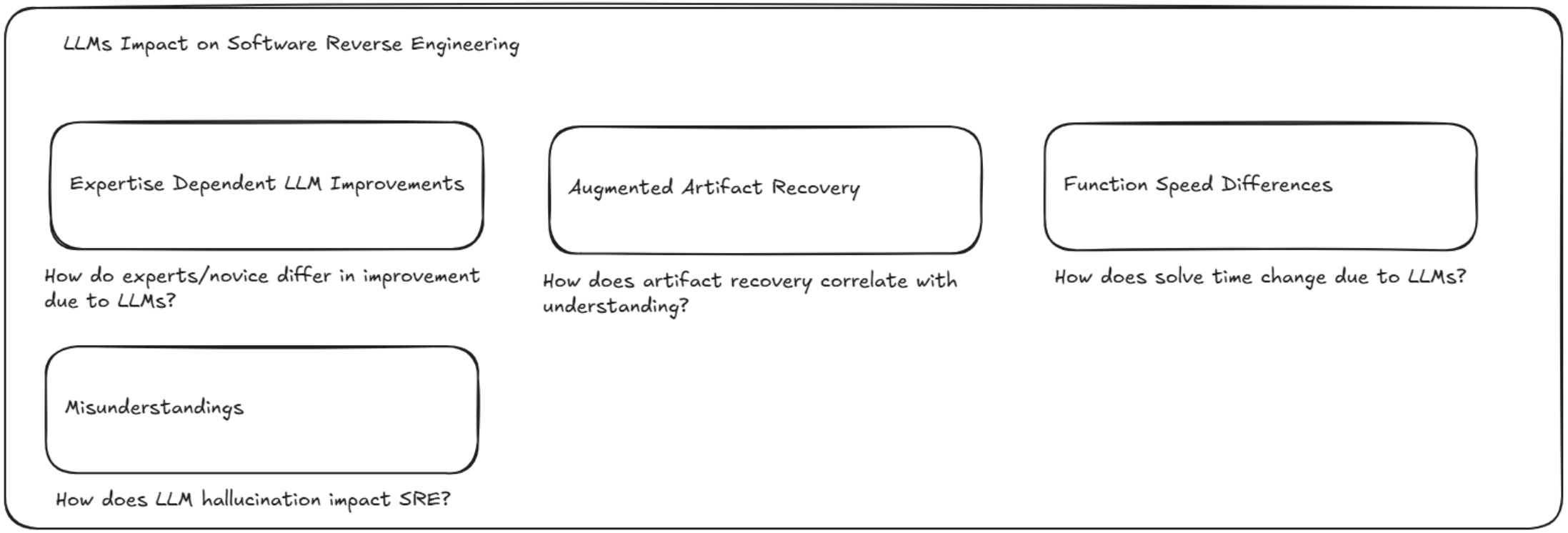
Six LLM Features identified.

How do you “measure” performance in SRE?

Manual assessment of participant’s writeup

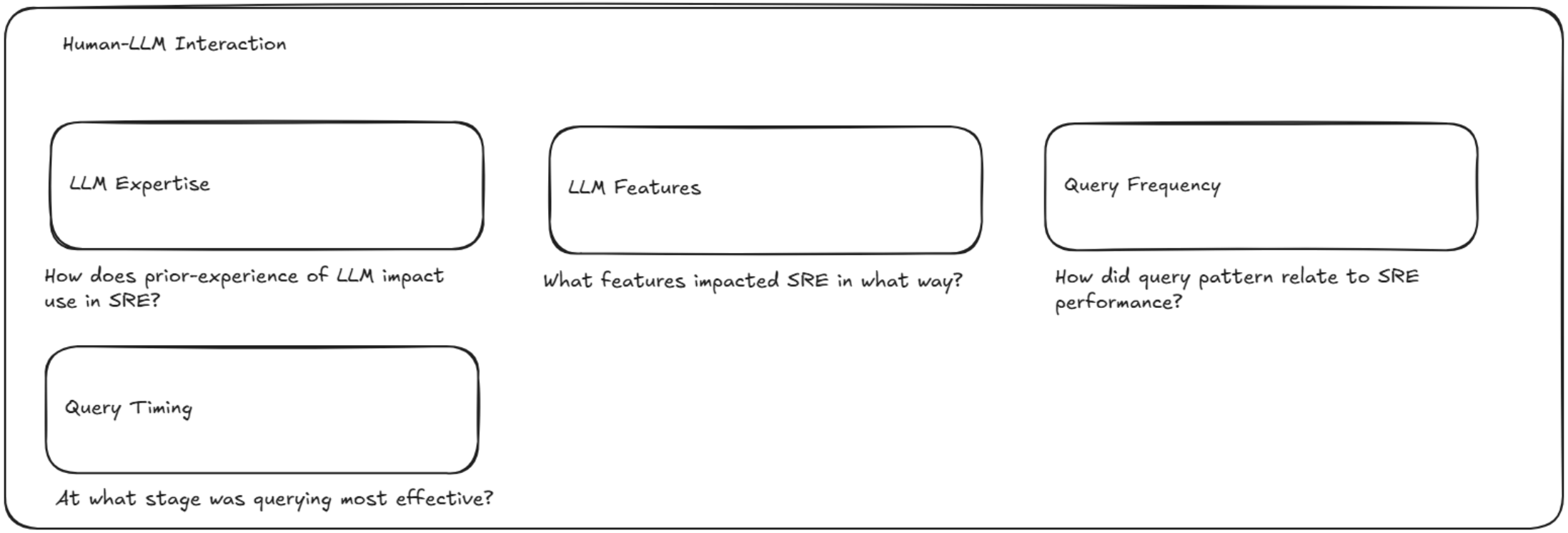
LLMs Impact on SRE (RQ2)

"How does the inclusion of LLMs in the SRE process impact the performance of practitioners?"



Human – LLM Interaction (RQ3)

"How do practitioners interact with LLMs, and what factors influence their interactions?"



We will not refer to all findings

- Total 18 Findings

- We will focus on **5 Key Takeaways** and relate back to the findings if needed.

1) LLMs Primarily shape the first moments of understanding, not deeper refinement

LLM shape first impression

- Early Impression determine the pace and quality of subsequent analysis (especially for novices)

Finding 18: "LLMs provide greater benefit when utilized at the beginning of function understanding, not the end"

Finding1: "Novices using LLMs exhibit expert levels of program understanding rates--regardless of whether experts use LLMs or not"

LLM shape first impression

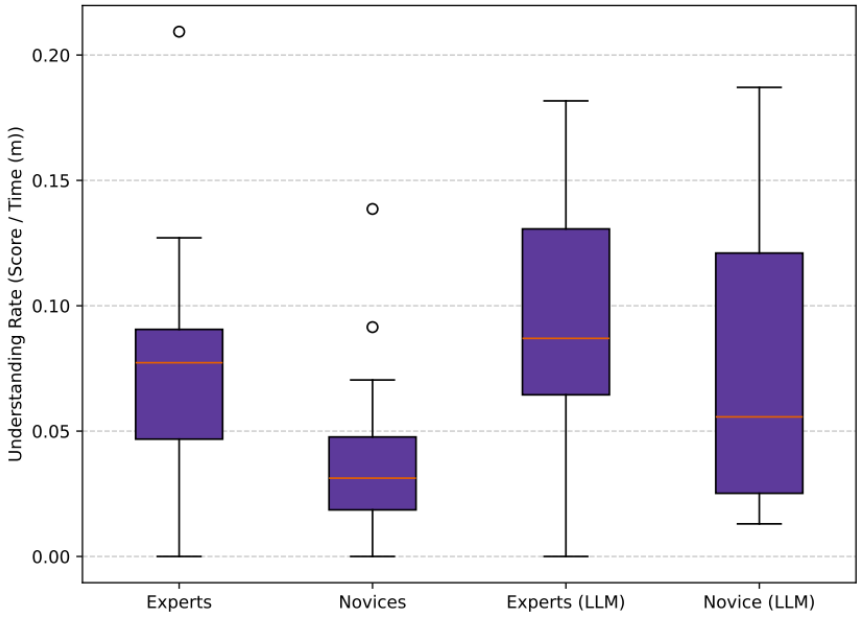


Figure 2: Understanding rates for participants with and without LLM assistance, grouped by SRE expertise. Higher understanding rates indicate higher performance SRE.

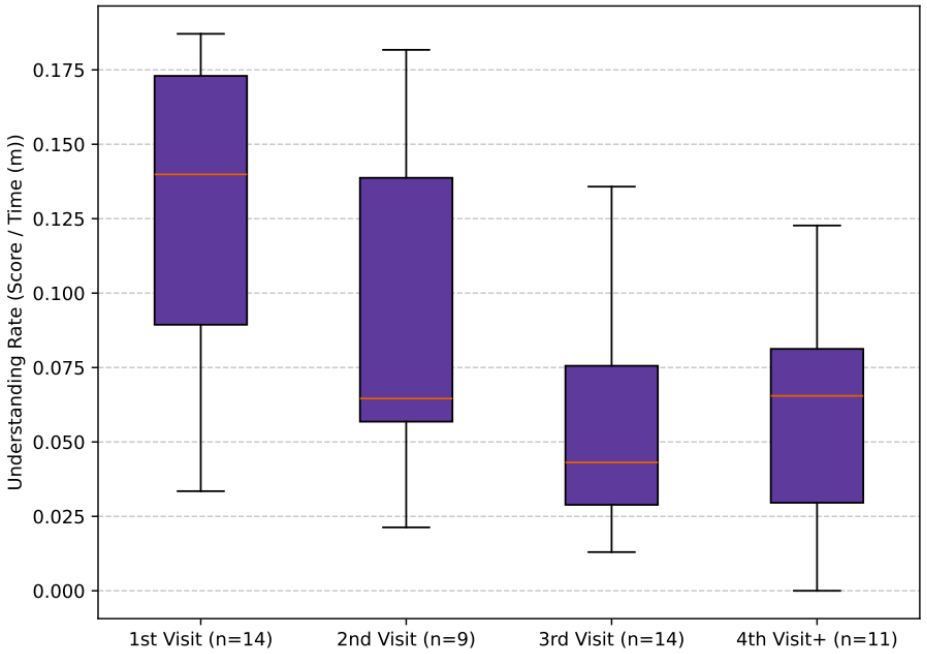


Figure 4: Understanding rates for participants that complete all queries on their first function visit vs on multiple visits.

LLM shape first impression but …

- Advantages do not extend to deeper or iterative understanding
- LLMs accelerate early semantic grounding but do not meaningfully assist with deeper analytical processes

Finding 16: "LLMs have degrading benefits when utilized repeatedly on individual functions"

Finding 2: "LLM usage does not impact experts' program understanding rate"

Finding 7: "Experts using LLMs spend less time on known algorithms and more time on custom ones"

2) The current interaction model of LLMs in SRE does not support expert knowledge refinement

Current LLMs do not adequately support experts

- Despite increasing adoption LLM-assisted SRE does not effectively support experts

> Finding 2: "LLM usage does not impact experts' program understanding rate"
>
> Finding 7: "Experts using LLMs spend less time on known algorithms and more time on custom ones"

- Without meaningful knowledge refinement, LLMs are likely to remain assistants for novices

3) Even rare hallucinations can severely derail SRE workflows

Rare hallucinations are still problematic

- Hallucination occurs rarely → their impact is disproportionally harmful

Finding 9: "LLM vulnerability hallucinations negatively impact subsequent human analysis"

- False vulnerability reports often disrupted participants
  - Pursue nonexistent flaws for extended periods

4) Semantic recovery often depends on the act of naming not merely the presence of names

Semantic recovery often **depends** on the **act of naming** not merely the presence of names

Relates a lot to what we do (something to think about)

Process itself is important!

- LLMs substantially increase the total number of recovered artifacts

> Findings 4: ""LLM artifact recovery is not correlated with improved program understanding"
>
> Findings 5: "LLM users recover more artifacts, including more false positives"

Process itself is important!

- LLMs substantially increase the total number of recovered artifacts

> Findings 4: ""LLM artifact recovery is not correlated with improved program understanding"
>
> Findings 5: "LLM users recover more artifacts, including more false positives"

- Artifact creation is itself an understanding process

> Finding 3: "Manual artifact recovery is positively correlated with program understanding"

Process itself is important!

- LLMs substantially increase the total number of recovered artifacts

  Findings 4: ""LLM artifact recovery is not correlated with improved program understanding"

  Findings 5: "LLM users recover more artifacts, including more false positives"

- Artifact creation is itself an understanding process

  Finding 3: "Manual artifact recovery is positively correlated with program understanding"

- LLMs are effective at generating “known algorithms”.
  - Relying on LLMs in complex contexts may hinder understanding

  Finding 6: "LLMs recover function names of known algorithms with a higher accuracy than humans"

5) Only certain forms of cognition can be effectively offloaded to LLMs

LLM effectiveness is limited to specific tasks

GOOD

- Summarizing well-scoped functions or identifying standard algorithms

  Finding 6: "LLMs recover function names of known algorithms with a higher accuracy than humans"

  Finding 18: "LLMs provide greater benefit when utilized at the beginning of function understanding, not the end"

BAD

- Summarizing functions that serve multiple roles, heavily optimized, or incorporate intertwined logic remain challenging

  Finding 17: "LLMs perform worse on larger functions (greater than 40 lines of code)

LLM effectiveness is limited to specific tasks

Top performers used LLMs less on larger functions than the bottom performers

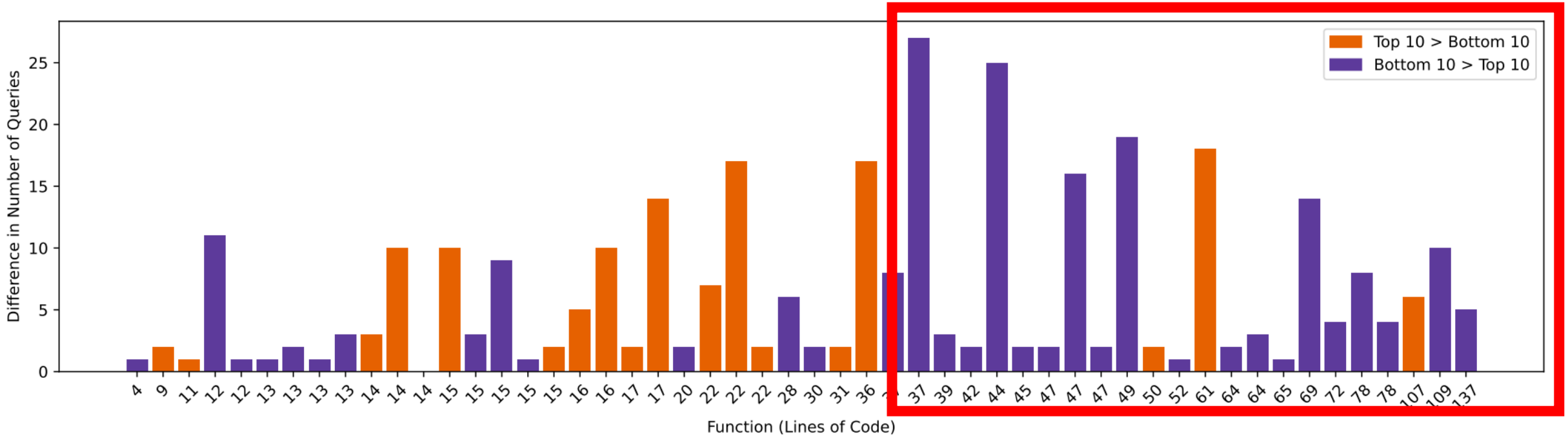


Figure 3: The difference in query amount by the top 10 and bottom 10 LLM users, sorted by function lines of code.

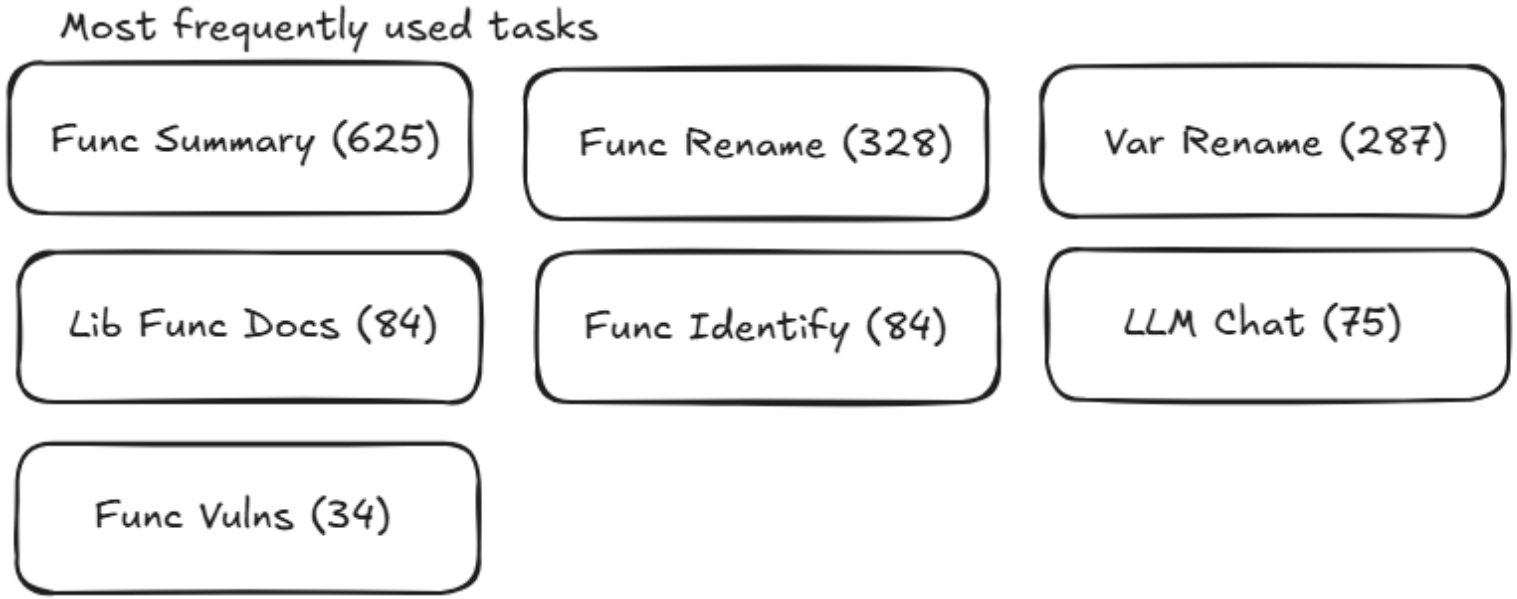
Other interesting findings

Finding 10: "Prior experience in using LLMs does not make them more useful for SRE"

Finding 11: "Experienced LLM users are more cautious about using LLMs"

Other interesting findings

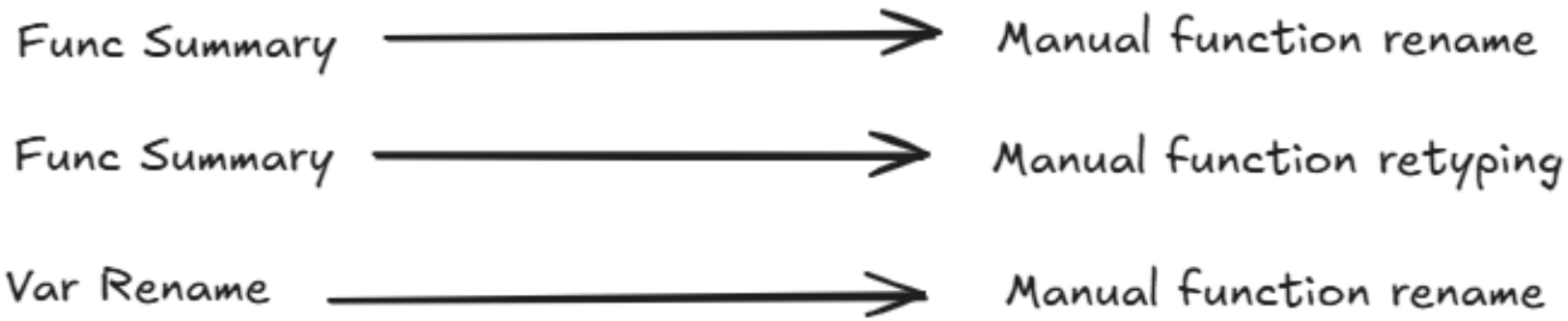
Finding 14: "LLM Func Summary and Var Rename lead to better program understanding"



Other interesting findings

Finding 14: "LLM Func Summary and Var Rename lead to better program understanding"

LLM features with best understanding rate



Takeaways

- This paper relates to lot of research we do in the lab.
  - You can use this paper as reference to support claims in the introduction
- If you are interested in human study this paper is good to check out
  - (e.g., how statistical significance was computed, how control/treatment groups were managed)
- A paper is not a gospel
  - Findings in this paper may not represent the entire SRE landscape (this is discussed in the paper as well)

The End.

Assumptions

- Self-reported SRE expertise is trustworthy

- Both challenges are of equal difficulty

Self Reported Expertise

Statistically Significant Difference
(p = 0.0001, es = 0.65, large effect size)

Expert: 6 Years (avg)
Novice: 2 Years (avg)

Statistical Difference
(p = 0.001, es = 0.56, large effect size)

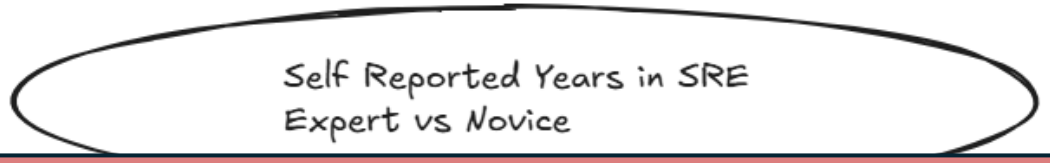
Expert: 4.79 /8 (avg)
Novice: 3.12 /8 (avg)

Statistical Difference
(p=0.001, es = 0.56, large effect size)

Expert: 0.08/min (avg)
Novice: 0.06/min (avg)

*p = p value, es = effect size
** Understanding rate = Score / solve time

Self Reported Expertise



Statistically Significant Difference
(p = 0.0001, es = 0.65, large effect size)

Expert: 6 Years (avg)
Novice: 2 Years (avg)

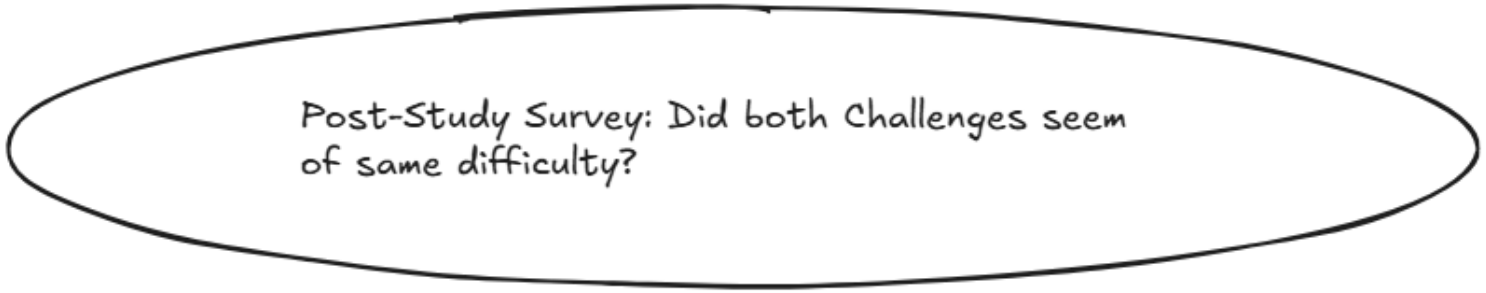
Statistically Speaking Self Reported Experts did show longer experience and better performance

Statistical Difference
(p=0.001, es = 0.56, large effect size)

Expert: 0.08/min (avg)
Novice: 0.06/min (avg)

*p = p value, es = effect size
** Understanding rate = Score / solve time

Equal Difficulty

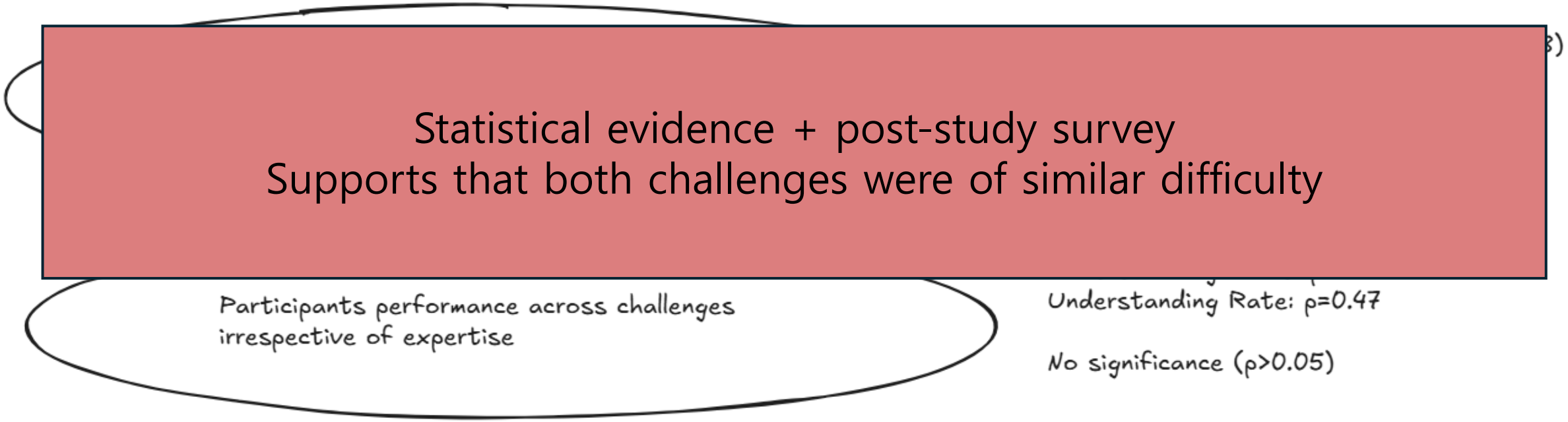


Strongly or Moderately Agree: 87% (43/48)
...
Strongly Disagree: 1% (1/48)

Understanding Score: $p=0.68$
Understanding Rate: $p=0.47$

No significance ($p>0.05$)

Equal Difficulty



Statistical evidence + post-study survey
Supports that both challenges were of similar difficulty