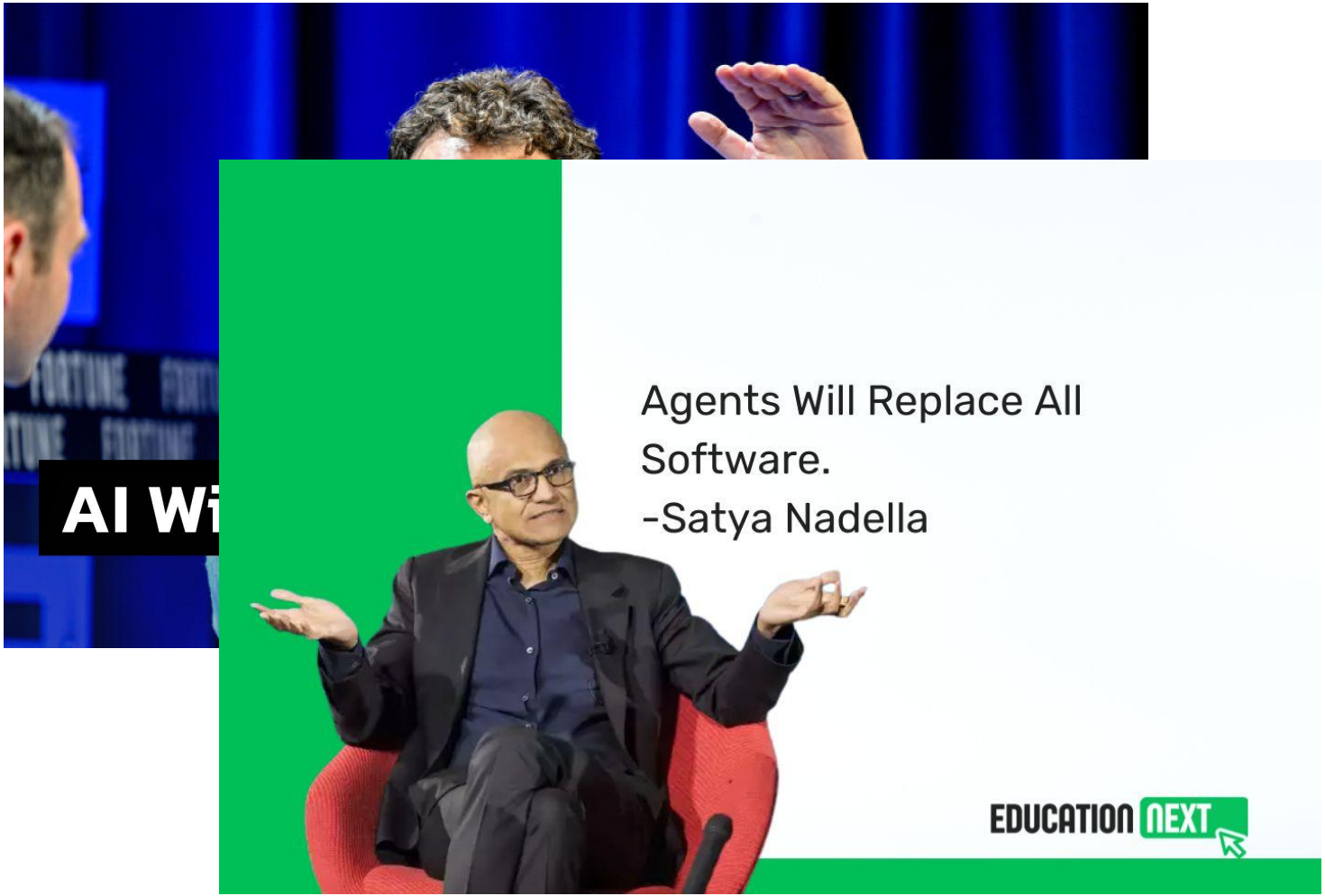
Success, sensitivity and unbelievable quality of LLM code generation

Andrei Paleyes and Diana Robinson

Artificial Intelligence Research Group Talks, Computer Lab
June 2025


AI Will Replace 90% Developers


AI Wi
Agents Will Replace All Software.
-Satya Nadella
EDUCATION NEXT

Kent Beck
@KentBeck

I've been reluctant to try ChatGPT. Today I got over that reluctance. Now I understand why I was reluctant.

The value of 90% of my skills just dropped to $0. The leverage for the remaining 10% went up 1000x. I need to recalibrate.

8:51 PM · Apr 18, 2023 · **1.4M** Views

End-user software engineering

“Because of these quality issues, researchers have begun to study end-user programming practices and invent new kinds of technologies that collaborate with end users to improve software quality. This research area is called end-user software engineering (EUSE).“

- Ko et al., 2011

Ko et al., “The State of the Art in End-User Software Engineering”, ACM Computing Surveys, 2011

End-user software engineering

“Our vision is that by 2030 end users will build and deploy whole apps just from natural requirements. We call this requirements-driven end-user software engineering.“

- Robinson et al., 2024

Robinson, Cabrera, Lawrence, Gordon, Mennen, “Requirements are All You Need: The Final Frontier for End-User Software Engineering”, International Workshop on Software Engineering 2030, 2024

I Am Devloper

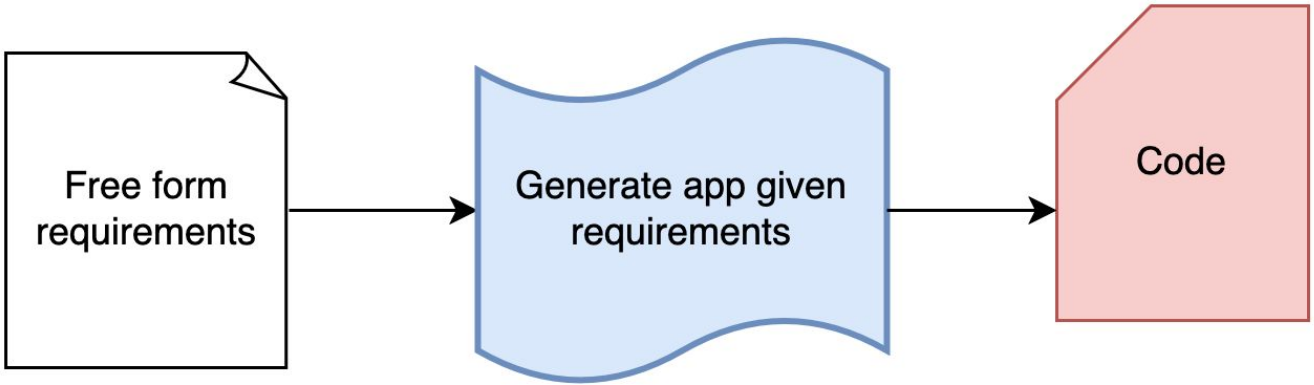
@iamdevloper

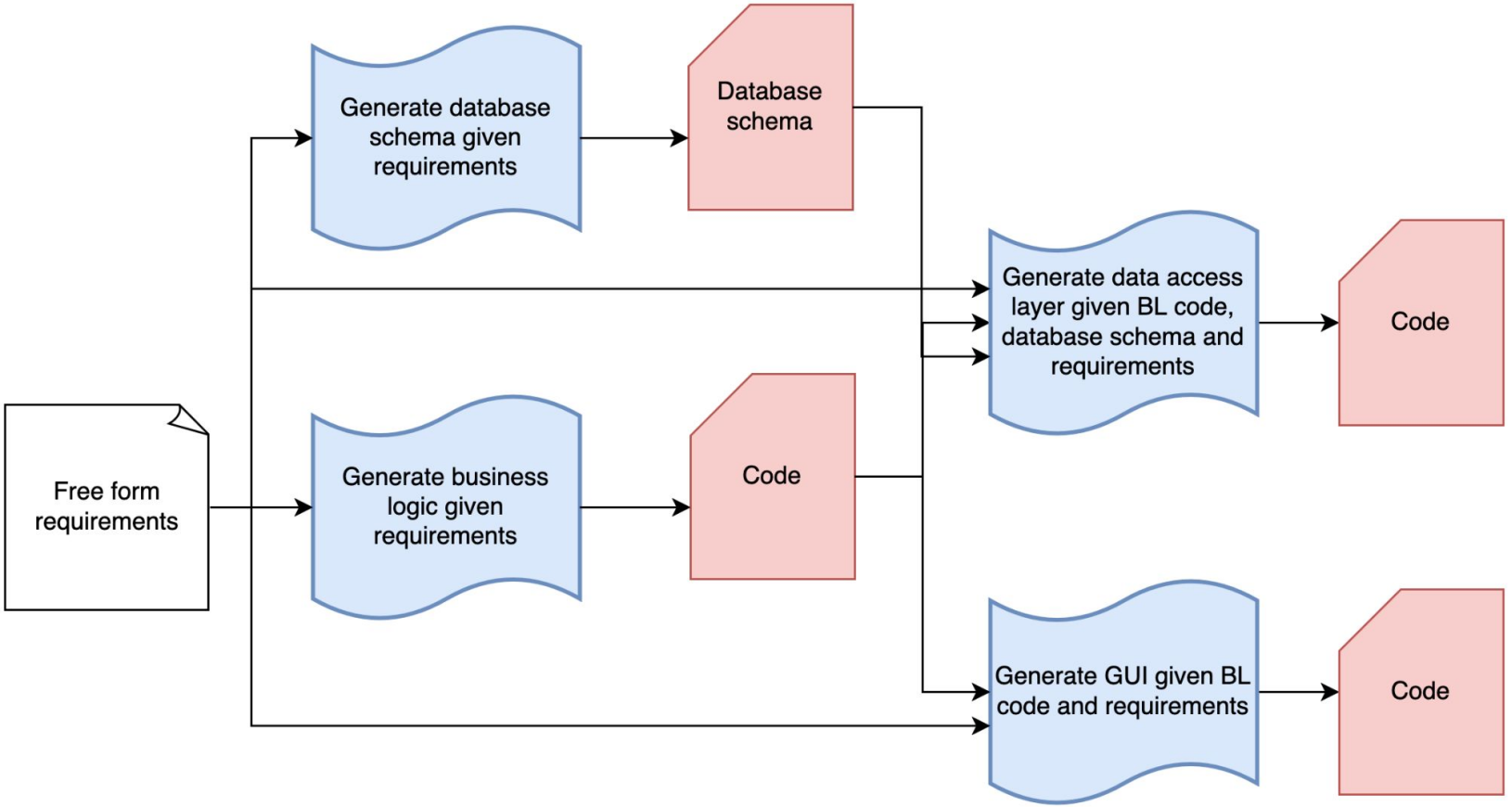
vibe coding, where 2 engineers can now create the tech debt of at least 50 engineers

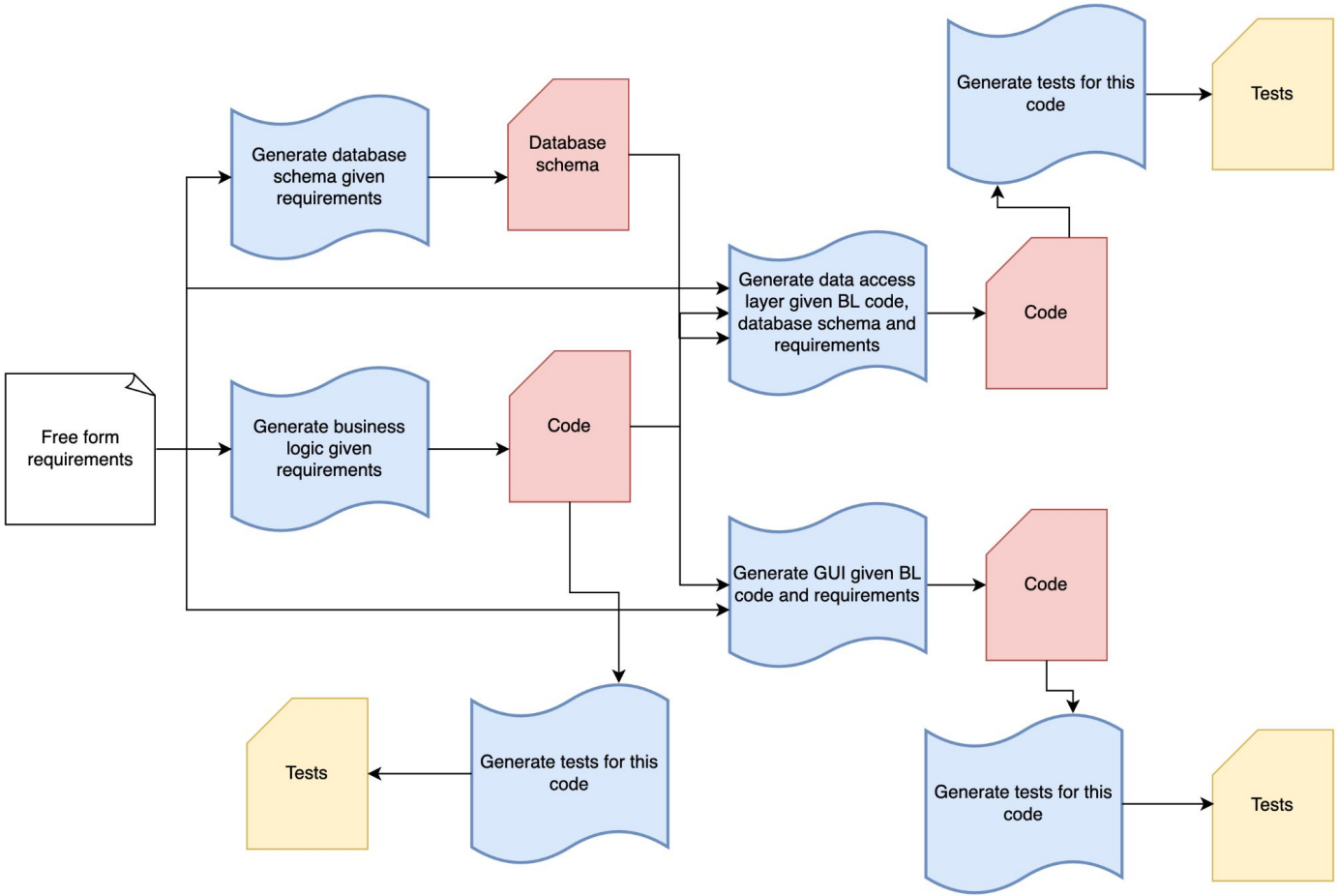
How to build reliable LLM coding pipelines?

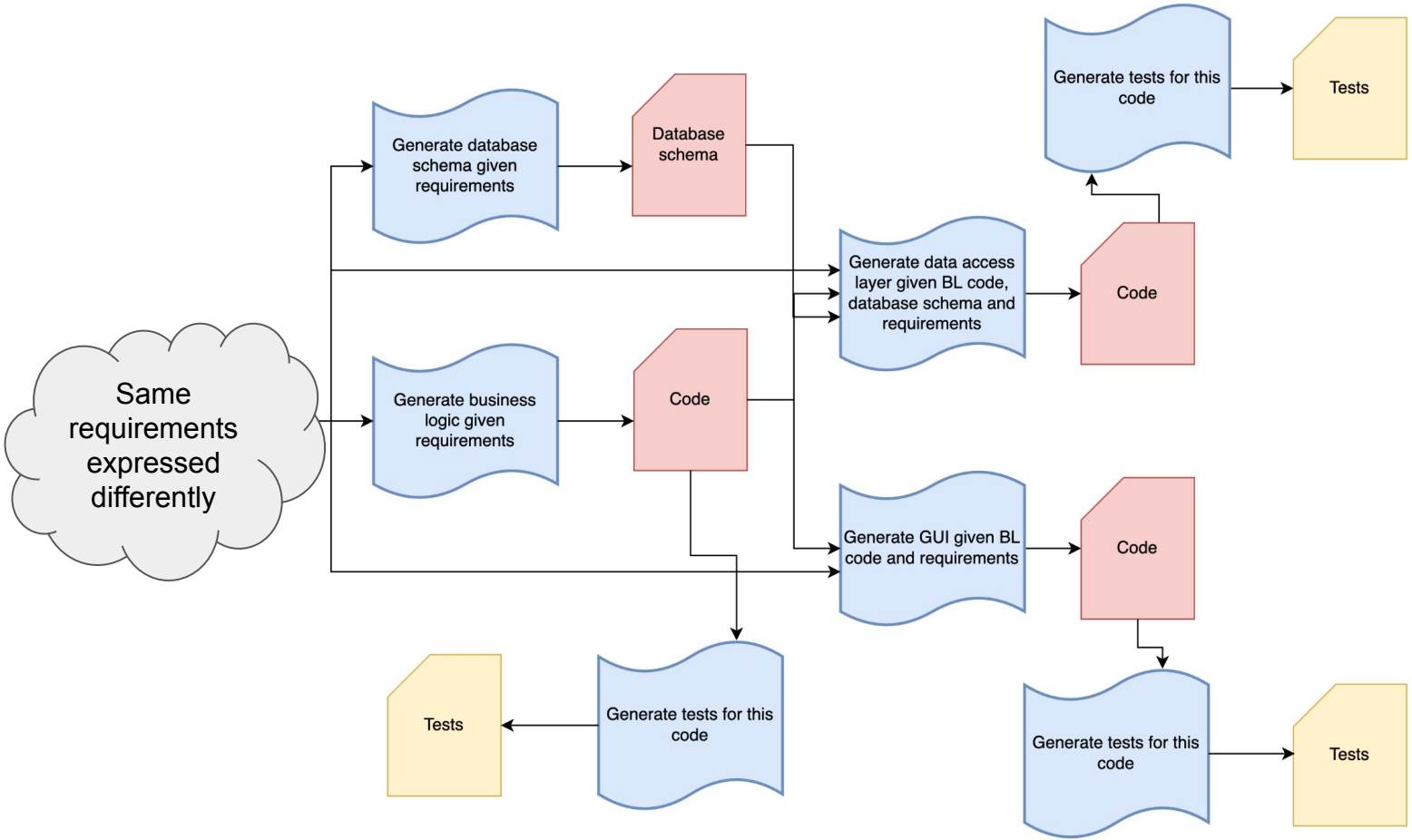
That’s one way…

And another, more solid one


Free form requirements
Generate app given requirements
Code

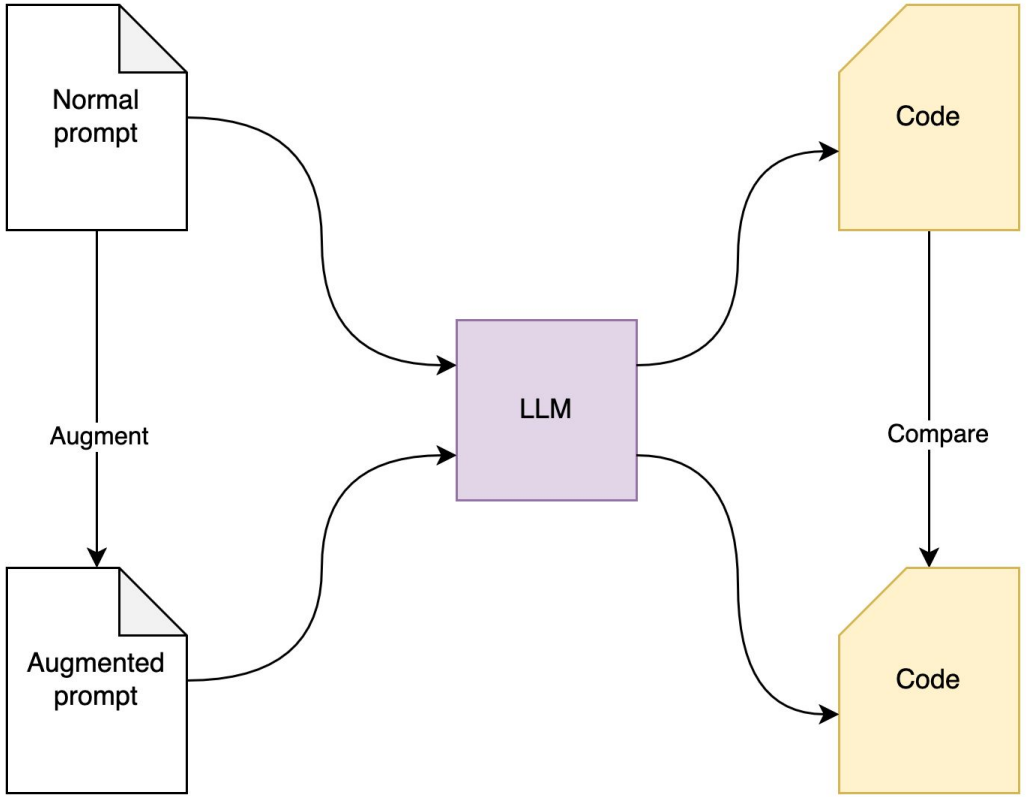

Free form requirements
Generate database schema given requirements
Database schema
Generate business logic given requirements
Code
Generate data access layer given BL code, database schema and requirements
Code
Generate GUI given BL code and requirements
Code


Free form requirements
Generate database schema given requirements
Database schema
Generate business logic given requirements
Code
Generate data access layer given BL code, database schema and requirements
Code
Generate GUI given BL code and requirements
Code
Generate tests for this code
Tests
Generate tests for this code
Tests
Generate tests for this code
Tests


Same requirements expressed differently
Generate database schema given requirements
Database schema
Generate business logic given requirements
Code
Generate data access layer given BL code, database schema and requirements
Code
Generate tests for this code
Tests
Generate GUI given BL code and requirements
Code
Generate tests for this code
Tests
Generate tests for this code
Tests

RQ: How sensitive is LLM code generation to variations in requirements?

Overview of evaluation



Augment

a → s

1. Typos

speak → talk

2. Synonyms

Write a program → Develop an app

3. Paraphrasing

As an English teacher with basic knowledge of computers…

4. Personas

Edward Ma, NLPaug, https://github.com/makcedward/nlpaug

Compare

❌Sacre BLEU

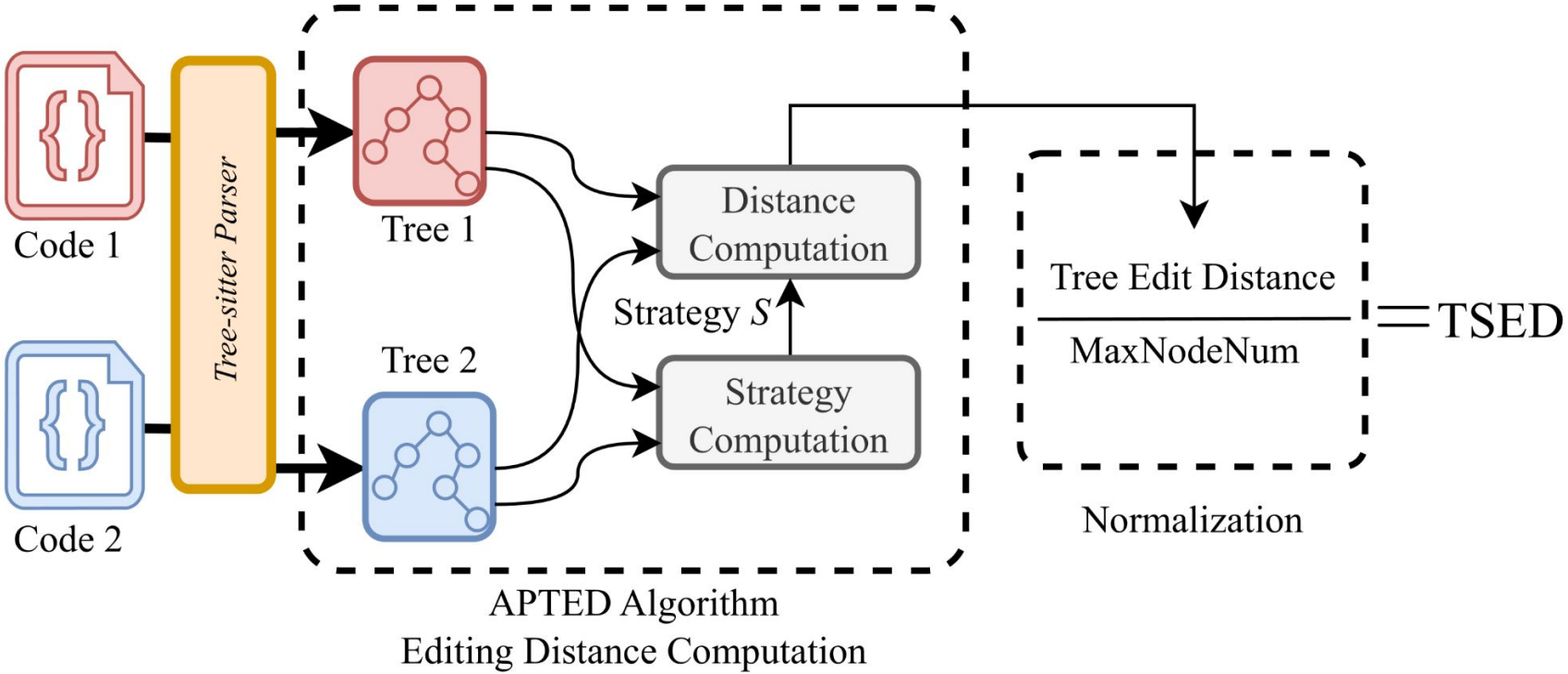
❌Jaccard similarity

❌BERT score

✔TSED

Evtikhiev et al., “Out of the Bleu: how should we assess quality of the code generation models?”, Journal of Systems and Software, 2023
Song et al., “Revisiting Code Similarity Evaluation with Abstract Syntax Tree Edit Distance”, ACL 2024

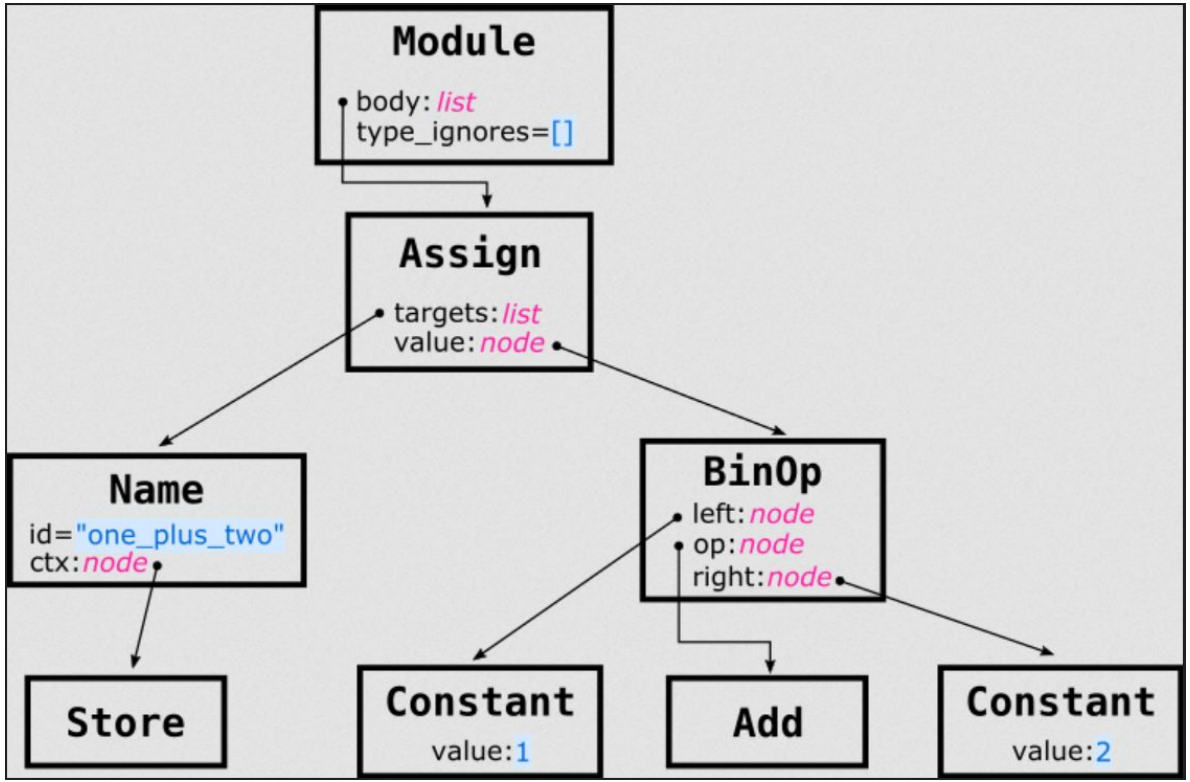
TSED = Tree Similarity of Edit Distance



Pawlik and Augsten, “Efficient computation of the tree edit distance”. ACM Transactions on Database Systems (TODS), 2015
Song et al., “Revisiting Code Similarity Evaluation with Abstract Syntax Tree Edit Distance”, ACL 2024

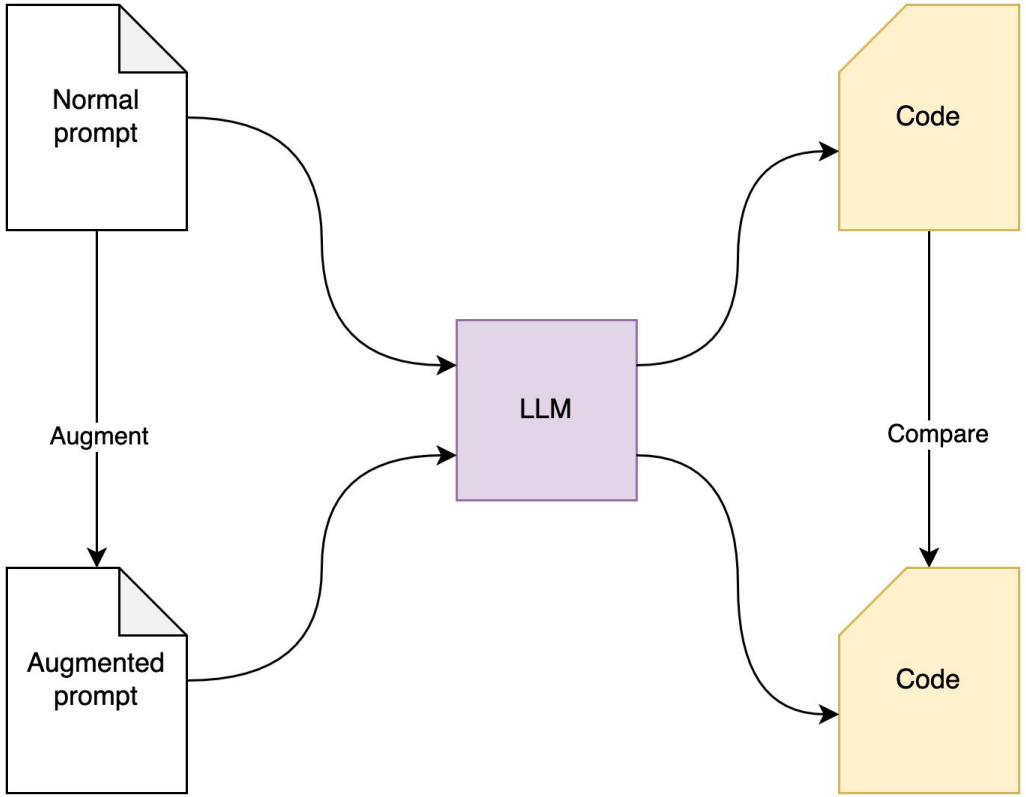
Abstract Syntax Tree

```
one_plus_two = 1+2
```

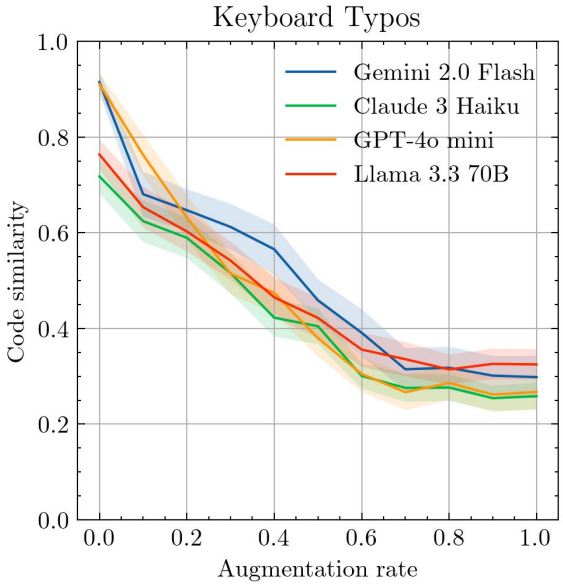


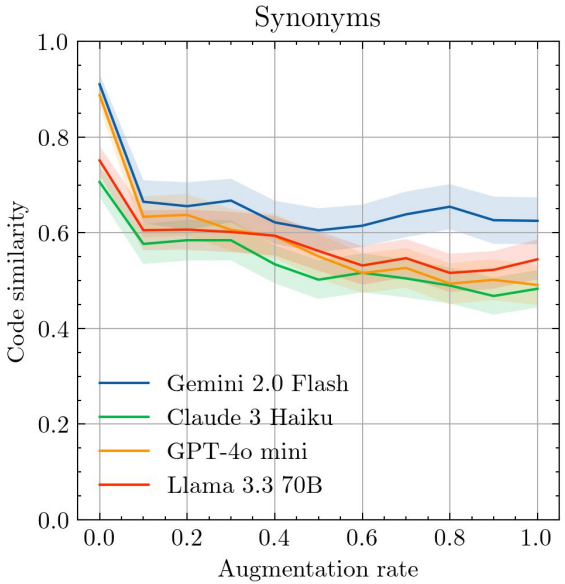
Alessandro Finamore, “Abstract Syntax Trees in Python”, https://pybit.es/articles/ast-intro/

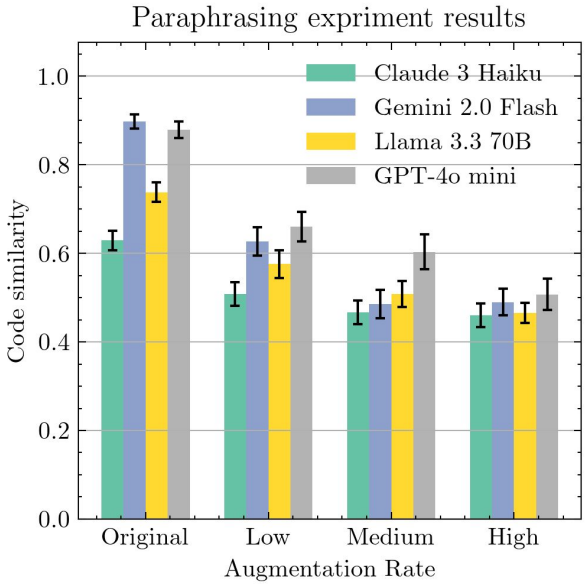
Overview of evaluation



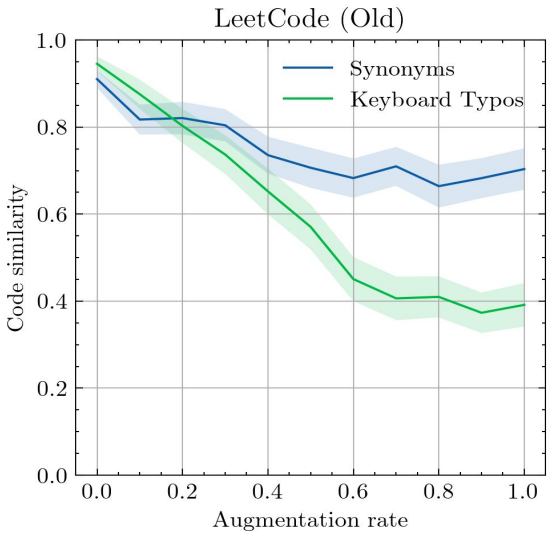
Results

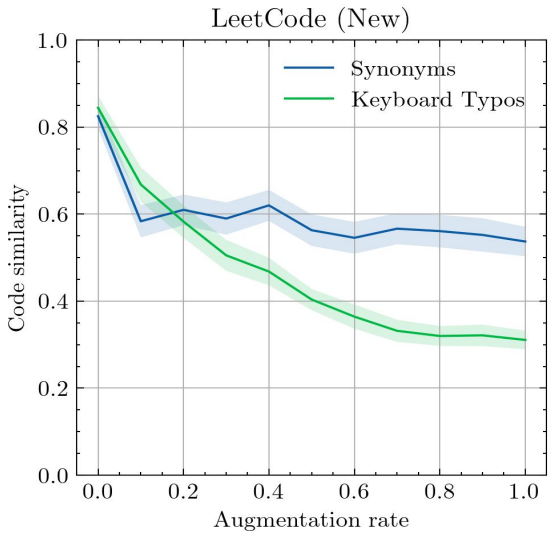

Keyboard Typos
Gemini 2.0 Flash
Claude 3 Haiku
GPT-4o mini
Llama 3.3 70B
Code similarity
Augmentation rate
0.0
0.2
0.4
0.6
0.8
1.0

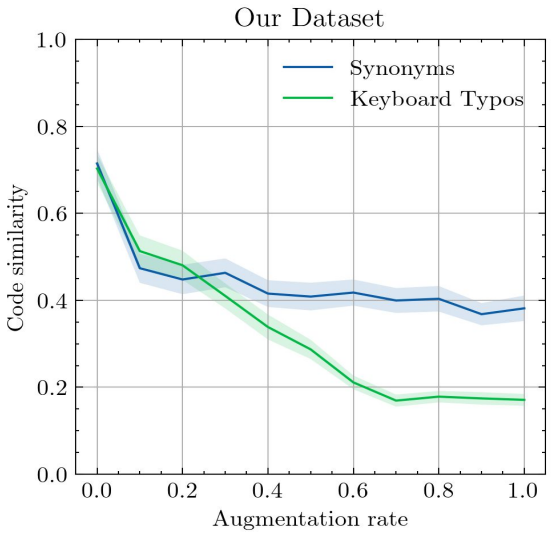

Synonyms
Gemini 2.0 Flash
Claude 3 Haiku
GPT-4o mini
Llama 3.3 70B
Code similarity
Augmentation rate
0.0
0.2
0.4
0.6
0.8
1.0


Paraphrasing expriment results
Claude 3 Haiku
Gemini 2.0 Flash
Llama 3.3 70B
GPT-4o mini
Code similarity
Original
Low
Medium
High
Augmentation Rate
0.0
0.2
0.4
0.6
0.8
1.0

Results


LeetCode (Old)
Synonyms
Keyboard Typos
Code similarity
Augmentation rate
0.0
0.2
0.4
0.6
0.8
1.0


LeetCode (New)
Synonyms
Keyboard Typos
Code similarity
Augmentation rate
0.0
0.2
0.4
0.6
0.8
1.0

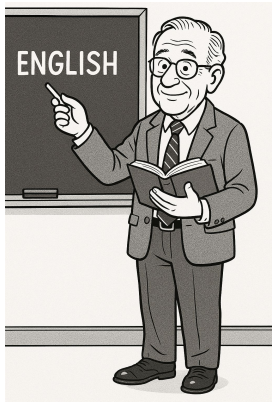

Our Dataset
Synonyms
Keyboard Typos
Code similarity
Augmentation rate
0.0
0.2
0.4
0.6
0.8
1.0

Personas - motivation

RQ 1: How does user background impact the way someone describes a prompt for a coding task?

RQ2: What difference does this type of prompt augmentation make to the generated code?

Personas - approach



Harold - english teacher

Ethan - astrophysicist

John - junior dev

Brad - principal dev

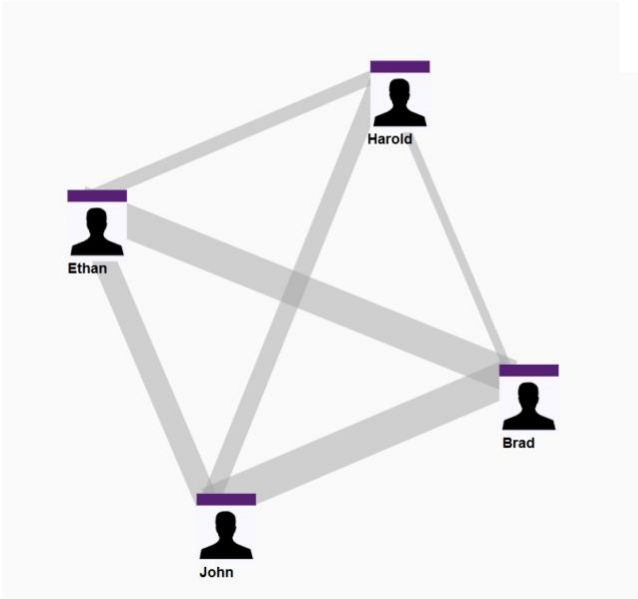
Image credit: ChatGPT, https://www.dreamstime.com/illustration/astrophysicist-cartoon.html, Perplexity

Personas – prompt difference results

- Example difference between English teacher persona and principal software developer:
    - “Can you provide me with a computer program that mimics a basic calculator?” (Harold)

      vs

    - “Create a simple web service with an API endpoint that performs calculations.” (Brad)



El-Assady et al., “Lingvis.io - a linguistic visual analytics framework”, ACL System Demonstrations, 2019

Personas - code difference results

- English teacher - basic code, lots of branching
- Astrophysicist - basic code, input and constraint validation
- Junior dev - OOP, use of databases
- Principal dev - data integrity, deployment considerations, KPI metrics

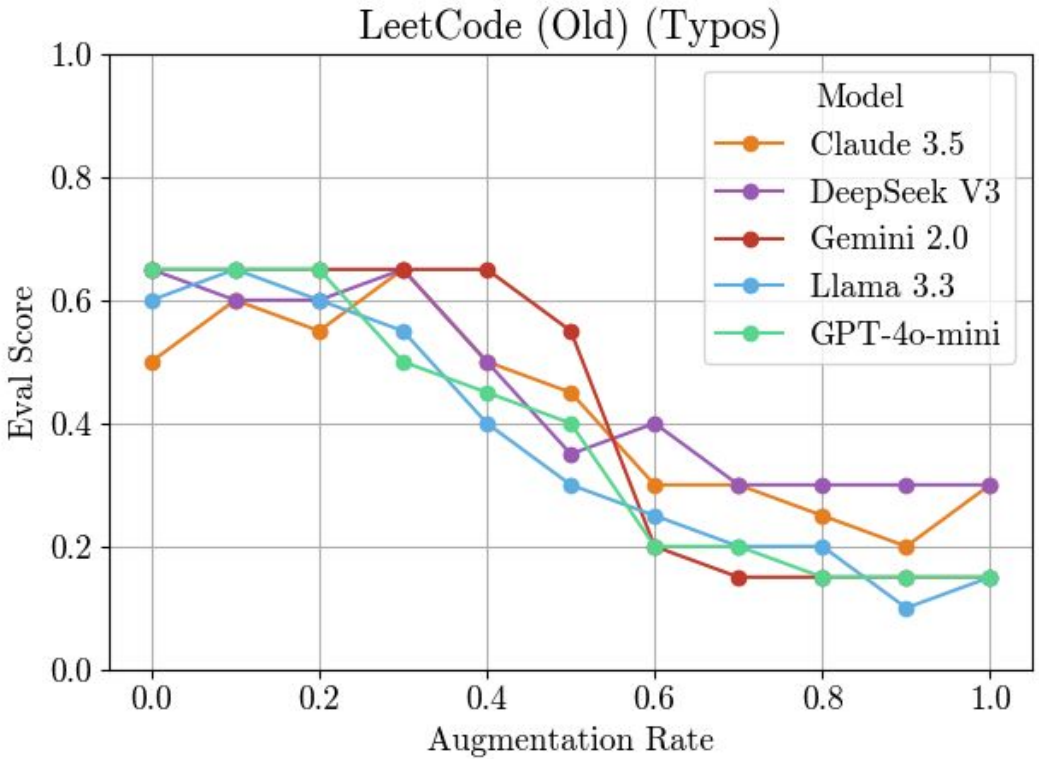
Aside: Claude writes ~3x more code than ChatGPT

LLMs are somewhat sensitive. So what?

- Full procedure
- Behavior is nuanced between LLMs
- Input data matters!...
- …as well as type of augmentation

P, Sendyka, Robinson, Cabrera, Lawrence “Prompt Variability Effects On LLM Code Generation”, arxiv:2506.10204, 2025

Aside on typos



Can you solve this?

Write Python code to solve the following problem:

hJFFm two a8Gh4E se5Zuz GIJe! and mj<Cq of q&Xs m and n %FE)2Xt(G#Oy, t$5 tDh the k$rJQH of the two EPEˆ@W xGeX %E. The (gSFq<: run F(K@ DkN(;ss9r7 W Bij>v be O (log (m + n)). K@jb$T = = n 0 <= m <= w000 0 <= n <= w000 1 <= m + n <= 1000 - 106 <= GHnZ@ [i], jk,e@ [i] <= 106

Can you solve this?

```
Given a signed 32-bit integ
```

Write Python code to solve the following problem:
hJFFm two a8Gh4E se5Zuz GlJe! and mj<Cq of q&Xs m and n %FE)2Xt(G#Oy, t$5 tDh the k$rJQH of the two EPEˆ@W xGeX %E. The (gSFq<: run F(K@ DkN(;ss9r7 W Bij>v be O (log (m + n)). K@jb$T = = n 0 <= m <= w000 0 <= n <= w000 1 <= m + n <= 1000 - 106 <= GHnZ@ [i], jk,e@ [i] <= 106

It looks like your prompt includes some garbled or obfuscated text, but I can deduce that you're asking for Python code to **find the median of two sorted arrays**, with a time complexity of **O(log (m + n))**, where:

- `0 <= m <= 1000`
- `0 <= n <= 1000`
- `1 <= m + n <= 1000`
- The arrays contain integers between `-10^6` and `10^6`

This is a classic problem. Here's the Python code to solve it using a **binary search approach**:

Given a signed 32-bit integ

It looks like your message got cut off. From what you've written — "Given a signed 32-bit integ..." — it sounds like you might be referring to a common coding interview problem such as:

> **"Given a signed 32-bit integer `x`, reverse its digits. If reversing `x` causes the value to go outside the signed 32-bit integer range, return 0."**

The **32-bit signed integer range** is from **-2^{31} to $2^{31} - 1$**, i.e., **–2,147,483,648 to 2,147,483,647**.

LLMs can match patterns. So what?

- Subtle changes can go unnoticed in automated pipelines
- Researchers: extra care when evaluating performance
- Developers: extra testing when synthesizing

Sendyka, P, Cabrera, Robinson, Lawrence, “LLM Performance for Code Generation on Noisy Tasks”, arxiv:2505.23598, 2025

Next steps

- Analyse actual pipelines, not only one-shots
- Personas → real humans
- Beyond code

Further:

- LLMs for software architecture
- LLMs for self-sustaining systems
- VibeSafe

Cabrera, Bastidas, Schooling, Lawrence, “The Systems Engineering Approach in Times of Large Language Models”, HICSS-58, 2024
Cabrera, P, Lawrence, “Self-sustaining Software Systems (S4): Towards Improved Interpretability and Adaptation”, SATrends, 2024

Team

Radzim Sendyka

Christian Cabrera

Diana Robinson

Neil Lawrence

AP

With thanks to

- Hongyu Zhou
- Andy Gordon
- Lars Mennen
- Viviana Bastidas
- Jennifer Schooling

Questions?

https://mlatcl.github.io/

ap2169@cam.ac.uk

dmpr3@cam.ac.uk