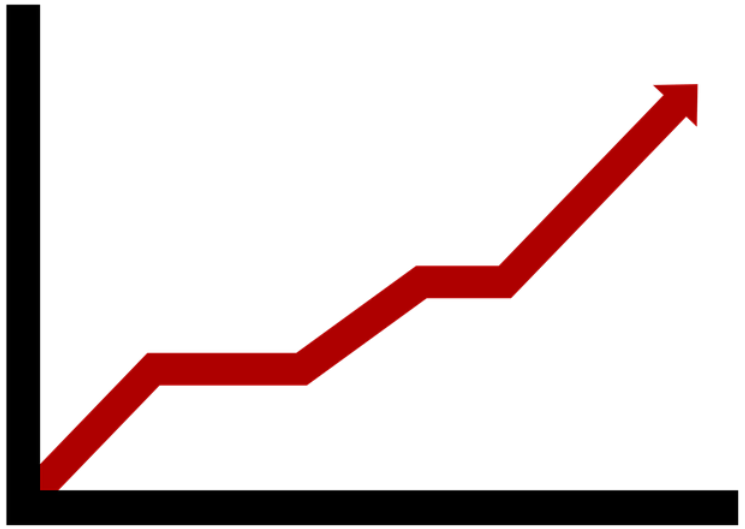
Evaluating the Quality of LLM-Generated Code

The Impact of LLMs



88% improvement in productivity[1]

Is this really true? **No!!**

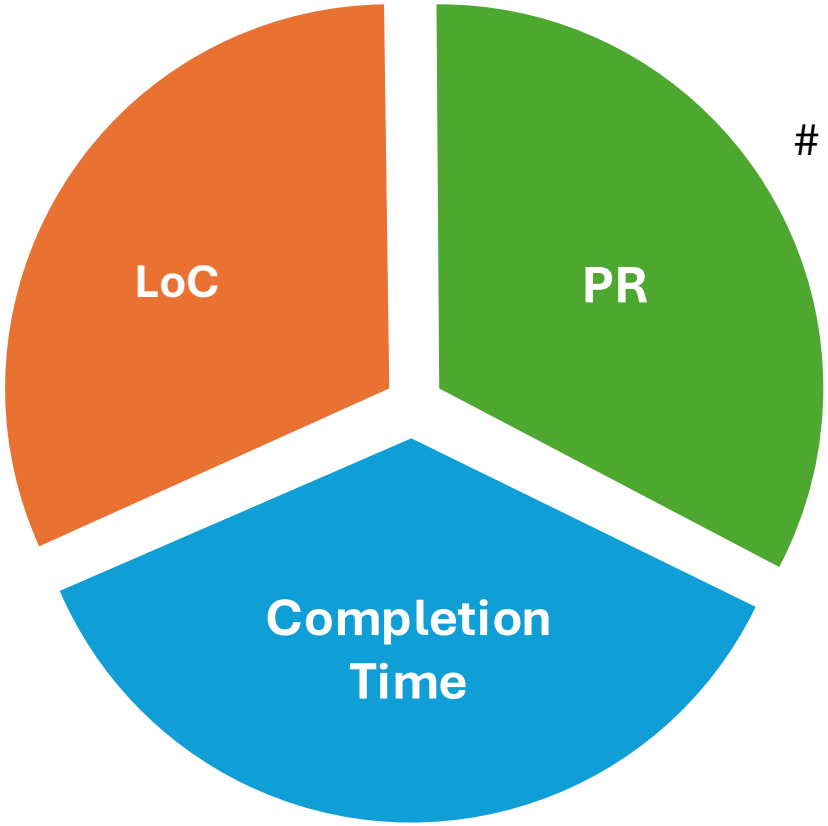
"Sooner than later, 80% of the code is going to be written by Copilot" GitHub CEO.[2]

1. Kalliamvakou, 2022
2. Scheffler, 2023

Misleading metrics

Industry's internal performance metrics

Lines of Code Produced

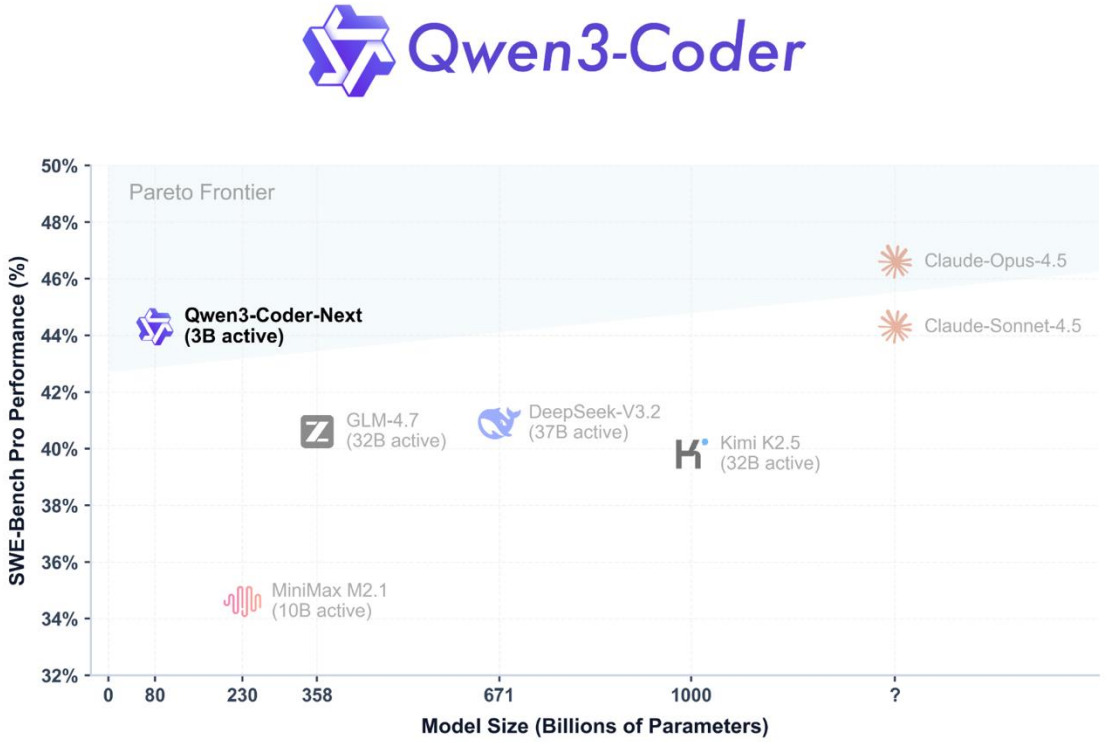


Pull Requests Created

Benchmark Overloading

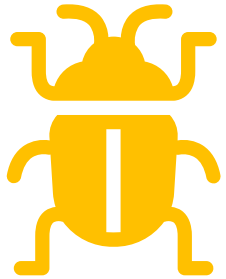
	Sonnet 4.6	Sonnet 4.5	Opus 4.6	Opus 4.5	Gemini 3 Pro	GPT-5.2 (all models)
Agentic terminal coding Terminal-Bench 2.0	59.1%	51.0%	**65.4%**	59.8%	56.2% (54.2% self-reported)	64.7% (64.0% self-reported) (Codex CLI)
Agentic coding SWE-bench Verified	79.6%	77.2%	80.8%	**80.9%**	78.0% (Flash)	80.0%
Agentic computer use OSWorld-Verified	72.5%	61.4%	**72.7%**	66.3%	—	38.2%
Agentic tool use τ2-bench	Retail 91.7% Telecom 97.9%	Retail 86.2% Telecom 98.0%	Retail **91.9%** Telecom **99.3%**	Retail 88.9% Telecom 98.2%	Retail 85.3% Telecom 98.0%	Retail 82.0% Telecom 98.7%
Scaled tool use MCP-Atlas	61.3%	43.8%	59.5%	**62.3%**	54.1%	60.6%
Agentic search BrowseComp	74.7%	43.9%	**84.0%**	67.8%	59.2% (Deep Research)	77.9% (Pro)
Multidisciplinary reasoning Humanity's Last Exam (HLE)	33.2% without tools 49.0% with tools	17.7% without tools 33.6% with tools	**40.0%** without tools **53.0%** with tools	30.8% without tools 43.4% with tools	37.5% without tools 45.8% with tools	36.6% without tools (Pro) 50.0% with tools (Pro)
Agentic financial analysis Finance Agent v1.1	**63.3%**	54.5%	60.1%	58.8%	55.2%	59.0%
Office tasks GDPval-AA Elo	**1633**	1276	1606	1416	1201	1462
Novel problem-solving ARC-AGI-2	58.3%	13.6%	**68.8%**	37.6%	31.1%	54.2% (Pro)
Graduate-level reasoning GPQA Diamond	89.9%	83.4%	91.3%	87.0%	91.9%	**93.2%** (Pro)
Visual reasoning MMMU-Pro	74.5% without tools 75.6% with tools	63.4% without tools 68.9% with tools	73.9% without tools 77.3% with tools	70.6% without tools 73.9% with tools	**81.0%** without tools — with tools	79.5% without tools **80.4%** with tools
Multilingual Q&A MMMLU	89.3%	89.5%	91.1%	90.8%	**91.8%**	89.6%

anthropic.com, 2026

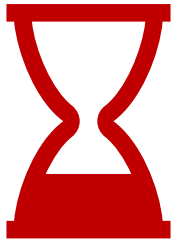


GitHub, Qwen3-Coder, 2026

Quality should be the main focus



40% of Copilot-generated code are buggy[3]



LLMs struggle to solve tasks that will take a human developer over 11 minutes to solve[4]

3. Pearce et al, 2022
4. Zhang et al, 2024

What do we mean by Quality?

- Does it solve the problem?
- Does the solution create other issues?
  - Bug-inducing fix

Problem Solving Skills of LLMs

Functional Correctness

Specific Input

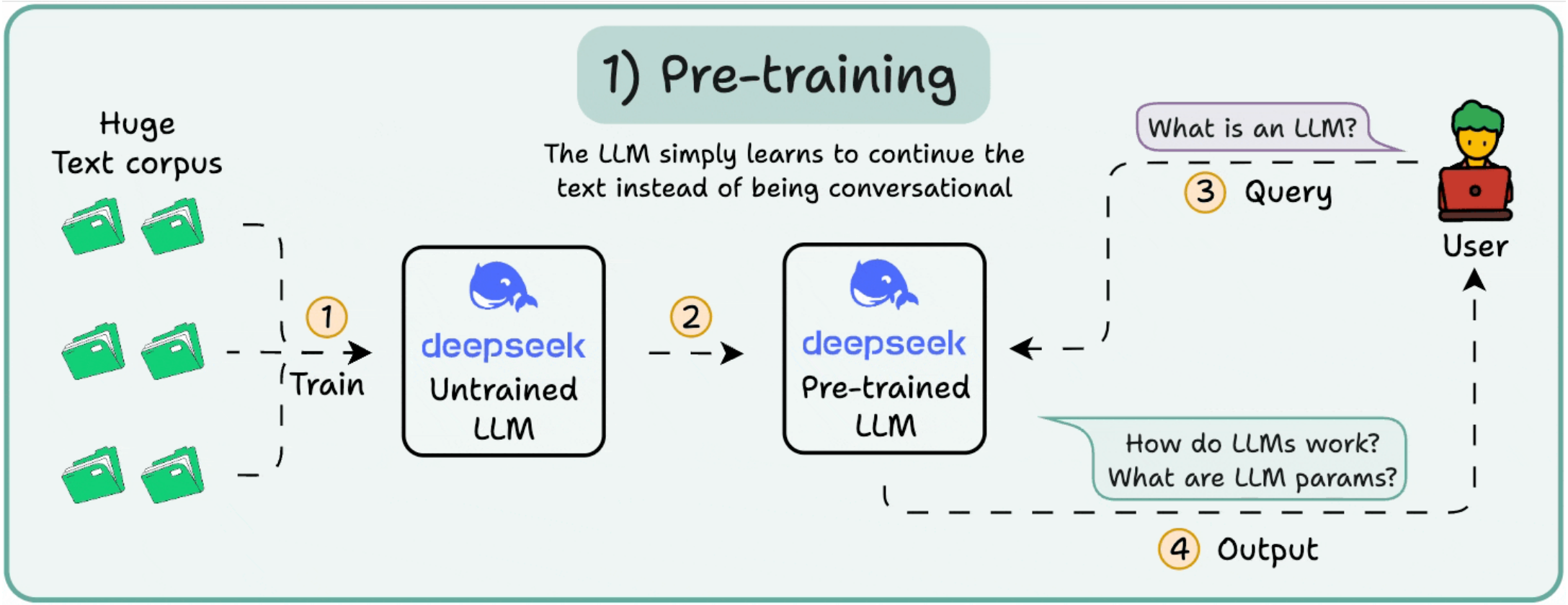
Specific Output

Functional Completeness

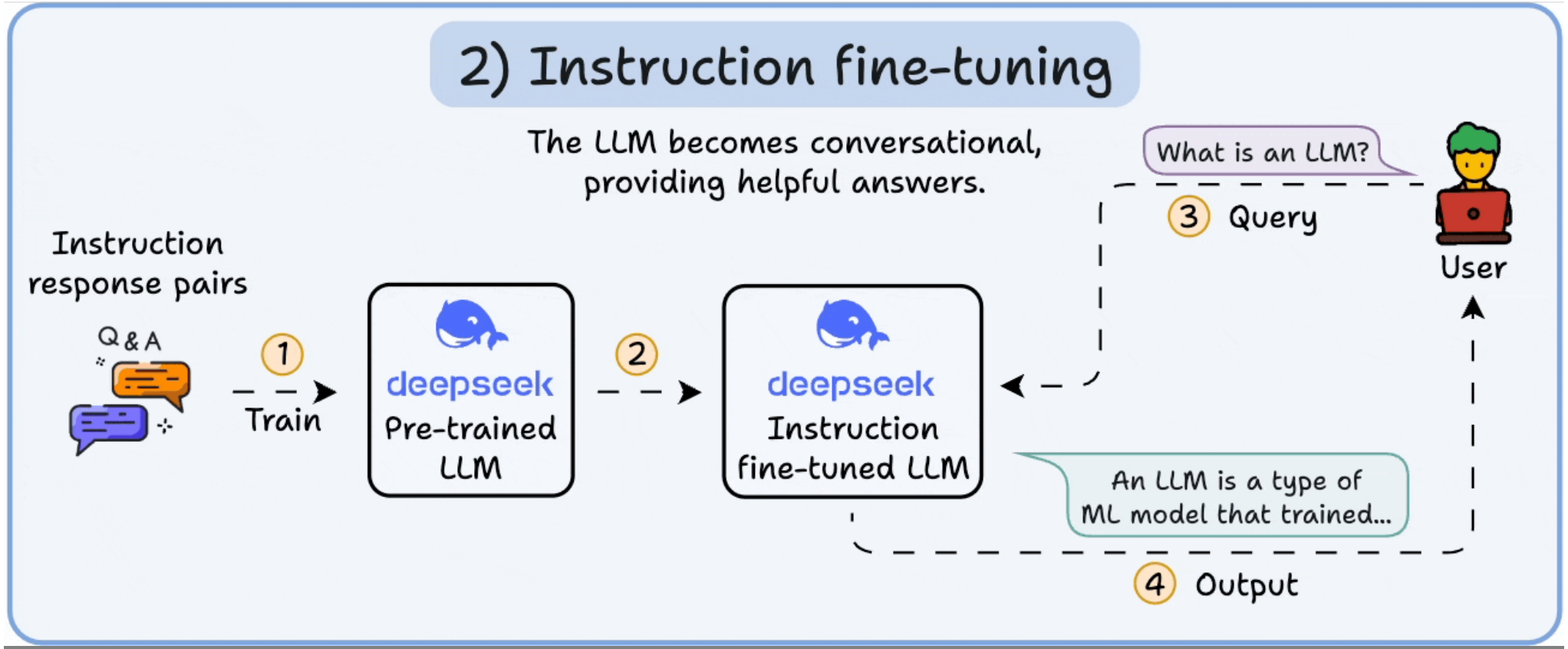
- Implement all the functionalities
- Fix all the bugs

How do LLMs work?

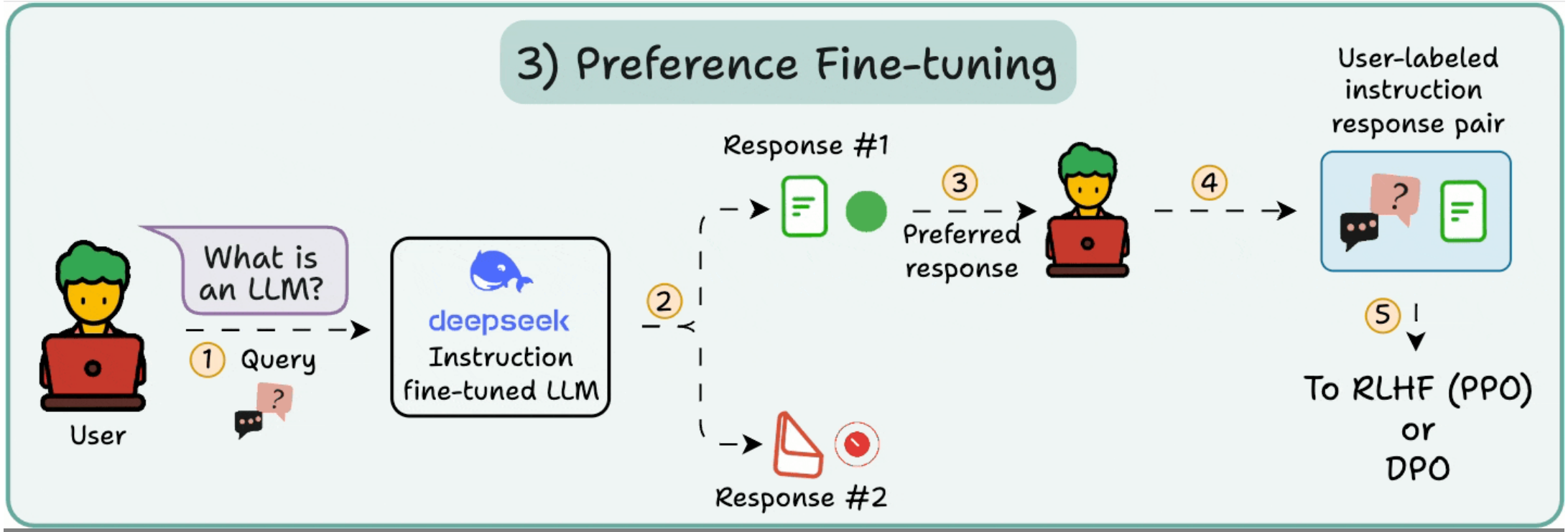
How do LLMs work?



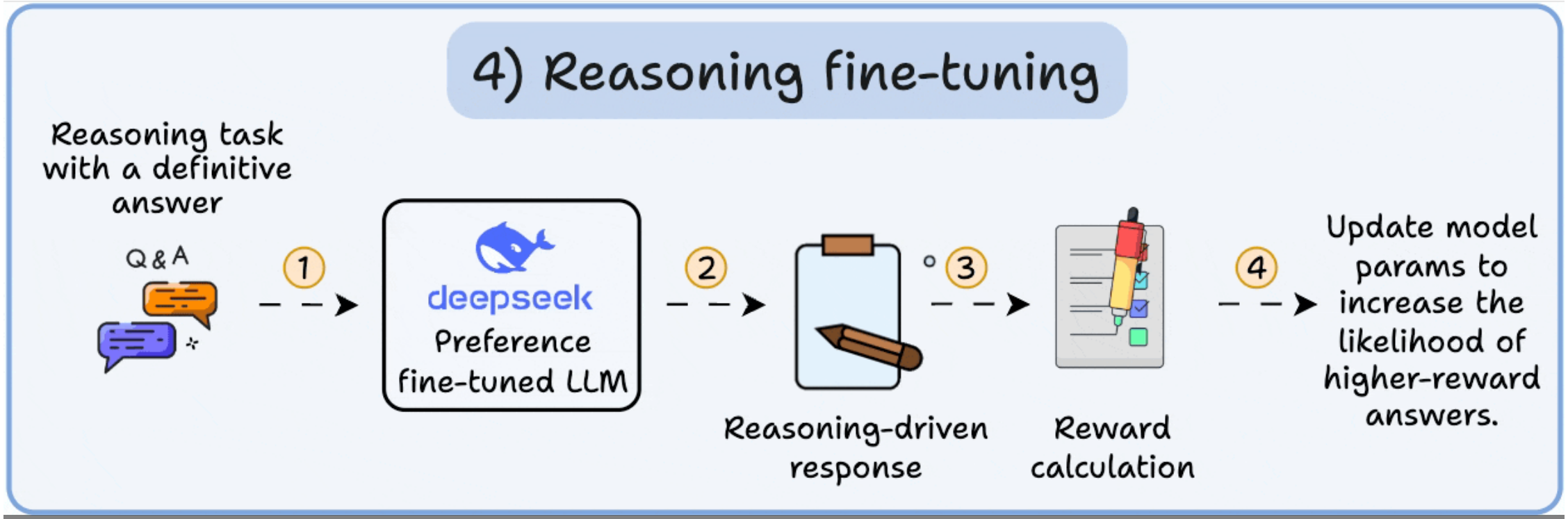
How do LLMs work?



How do LLMs work?



How do LLMs work?



Trained data

- LeetCode
  - Small tasks
  - Does not represent actual software development tasks
- Benchmark
  - LLMs are trained on this, usually
- Data structures, Algorithms, etc.
- Web Applications
- Boilerplate code

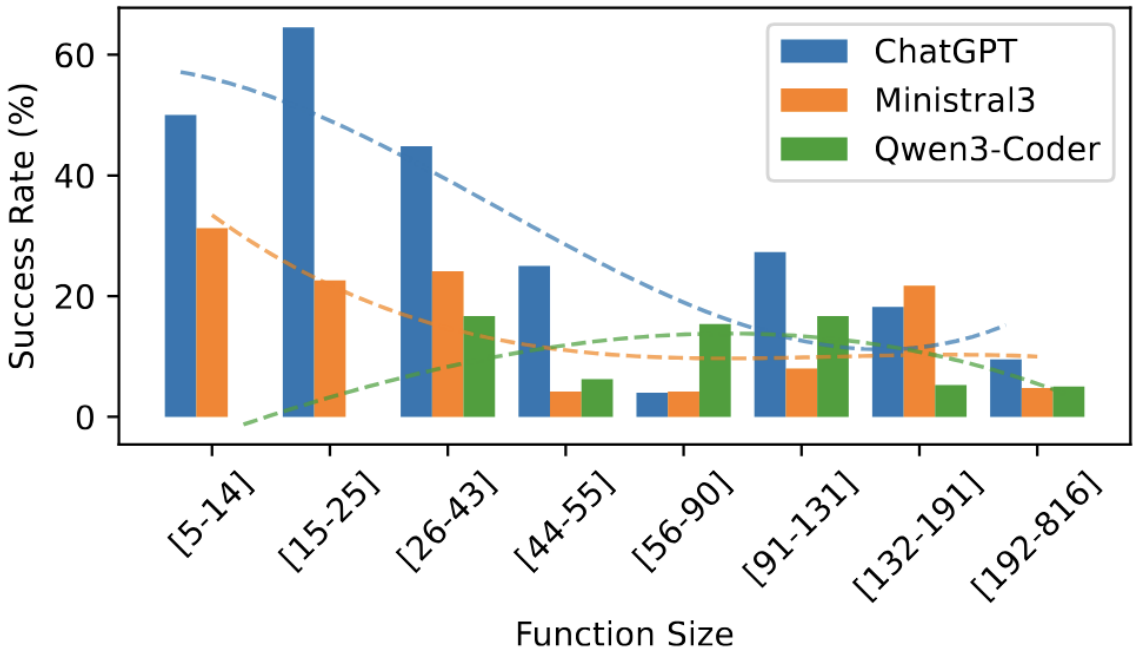
Other contributing factors to the success rate of code generation

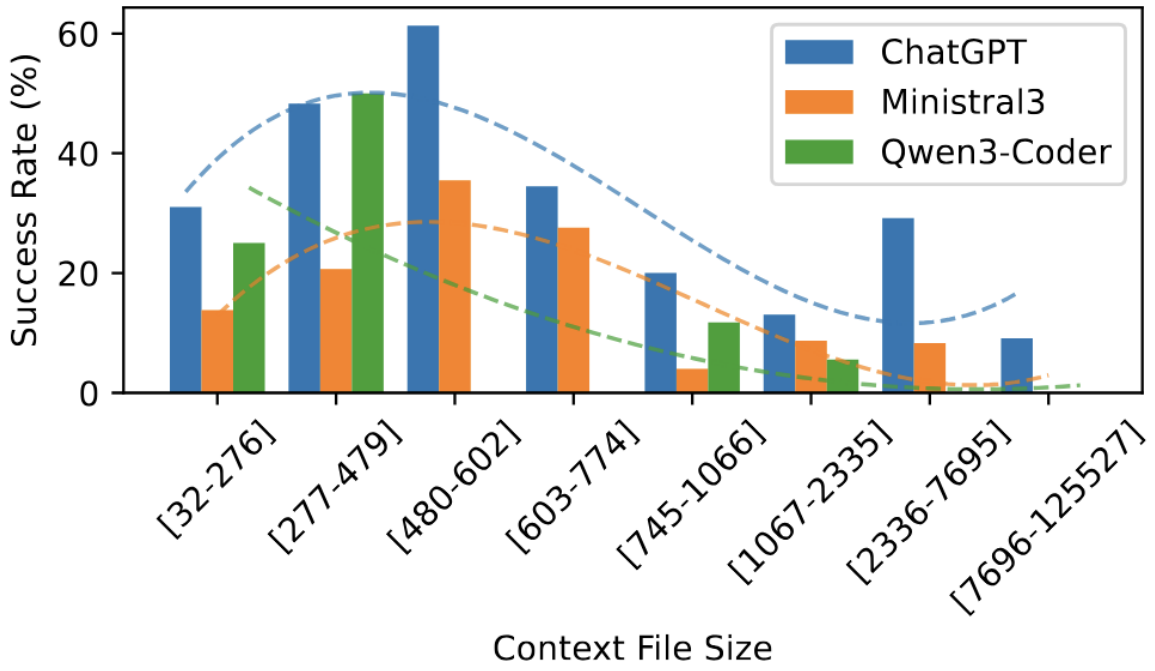
- Function size
- Context size

Nature of the tasks

- Bug-fixing tasks
- New feature implementation

Function Size, Context Size





Nature of the tasks

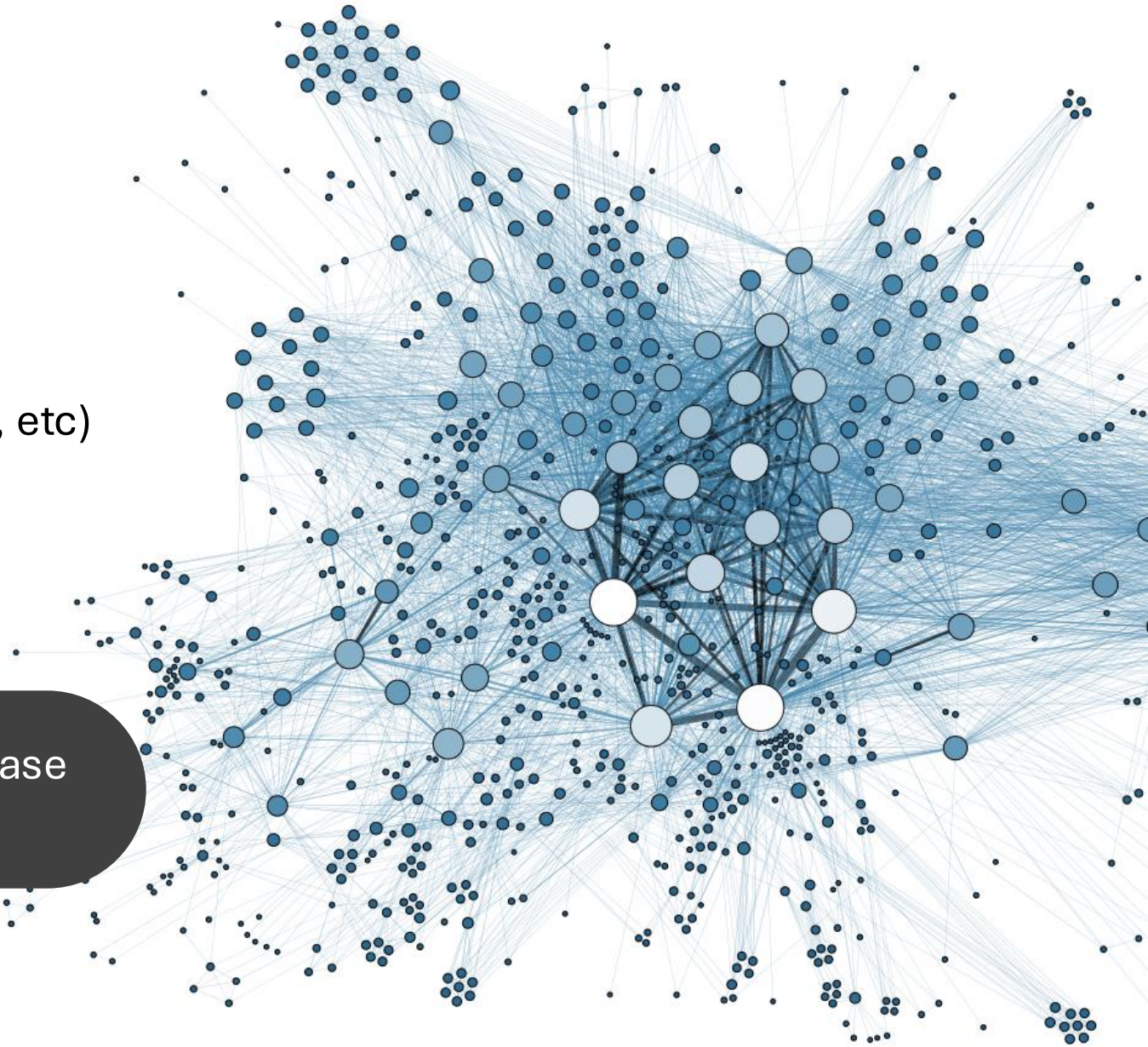
	Feature Enhancement	Bug Fix
Correct	13	56
Wrong	12	131
Success Rate	*52%*	*29.9%*

Unseen Data

Code obfuscation on existing dataset

- Alpha renaming (variable, function names, etc)
- Changing the control flow of the program (unnecessary ifs, for loop, jump, etc)
- Adding garbage code

Average passing rate of test cases can decrease up to 62.5%[5]



5. Zhang et al, 2025

An example of code obfuscation

```
private boolean isCS485Awesome ( ) {
  if (2 % 2 == 0) {
    return true;
  }
}
```

```
private boolean isCS485Awesome ( ) {

  if (isEven(2)) {
    int iIsActuallyAlooooooopcounterrrrrrrr = 0;

    while (iIsActuallyAlooooooopcounterrrrrrrr < 10) {

      if (iIsActuallyAlooooooopcounterrrrrrrr % 2 == 0) {
        System.out.println("Professor Kellogg is awesome.");
      }
      else {
        System.out.println("But TA is aite.”);
      }
    }
    return true;
  }
  else {
    return false;
  }
}

private boolean isEven (int num) {
  if (num % 2 == 0) {
    return true;
  }
  else {
    return false;
  }
}
```

Unseen Data

Commits on Open-Source projects that came after Knowledge-Cutoff date.

		Success Rate	
		LOC<Median	LOC>Median
Knowledge Cutoff Date	Before	17.4% (4 out of 23)	13.6% (3 out of 22)
	After	14.3% (2 out of 14)	6.3% (1 out of 16)

A correct solution is not the end

Hallucination[6]

Create irrelevant and complicated conditional logic[7]

Bug inducing fix

Misuse of APIs[8]

- Custom memory allocation/data structure in C
- Custom data structure/object in Java

Lacks defensive programming[9]

- Checks for NULL
- Integer overflow

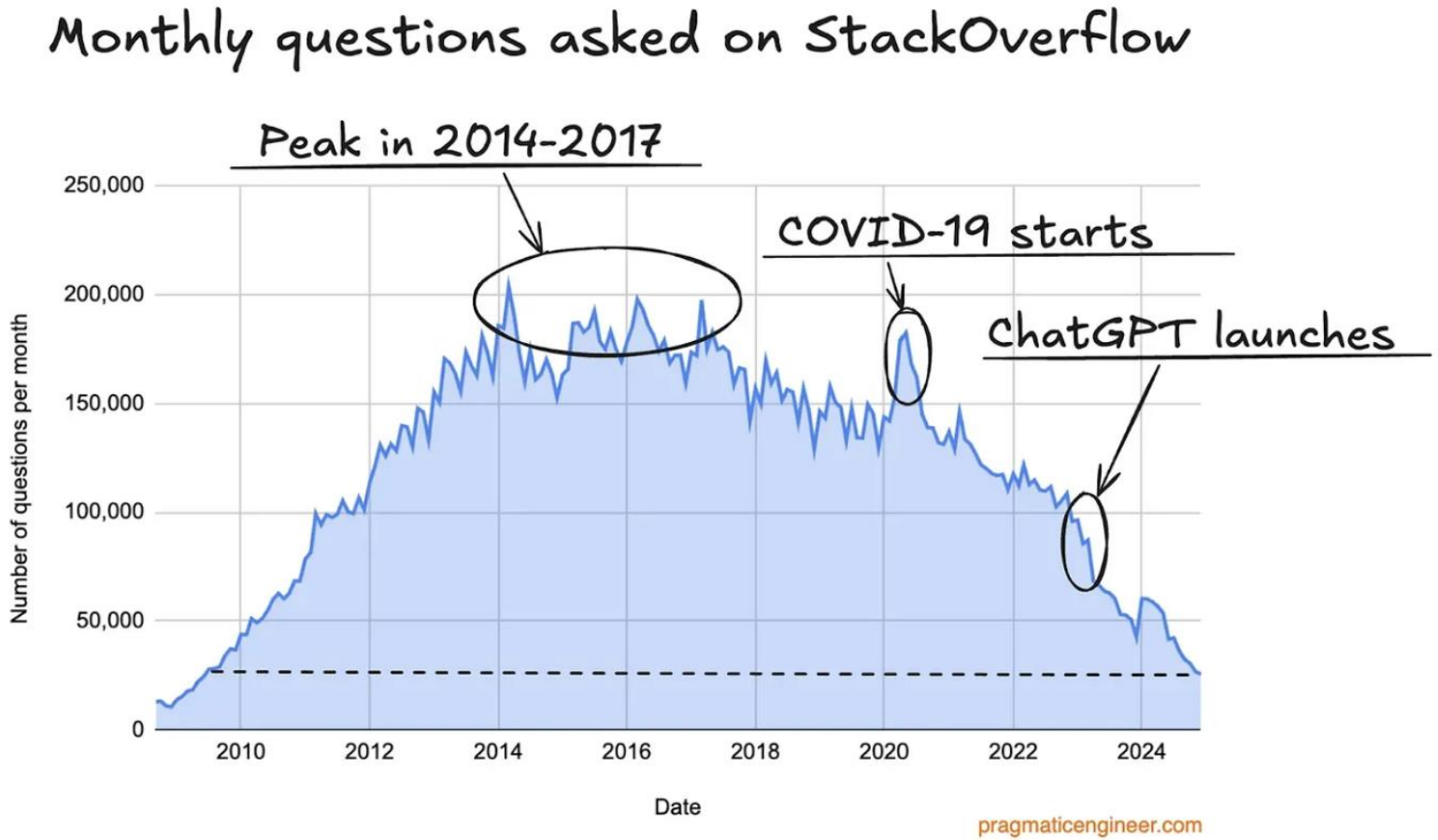
6. Liu et al, 2024
7. Li et al, 2024
8. Chen et al, 2025
9. Chong et al, 2026

Is AI Generated Code Considered Harmful?

AI is just a tool

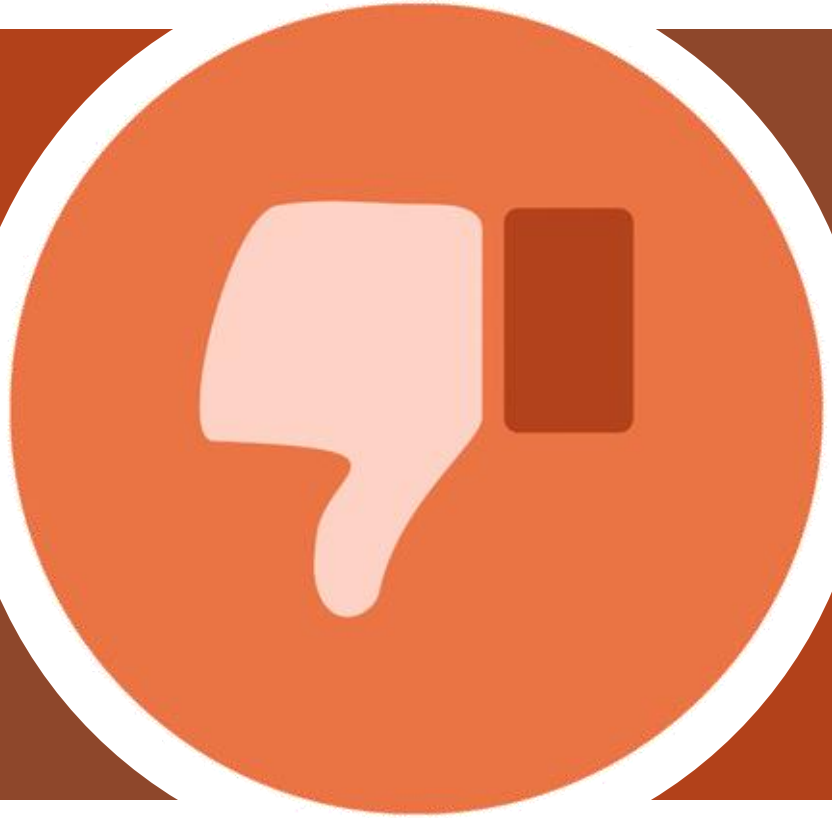
- Don’t romanticize it
- A means to an end, but not the end.
- Random number generation from GPT. (42)
- Google search (or StackOverflow) on steroids

StackOverflow



AI is harmful when

You become a tool of AI



We check if you check the output of AI in this class

You should be the last guardrail of AI

We check if you understand what you're prompting

In fact, AI is really harmful because…

MENU ≡ TIME SUBSCRIBE

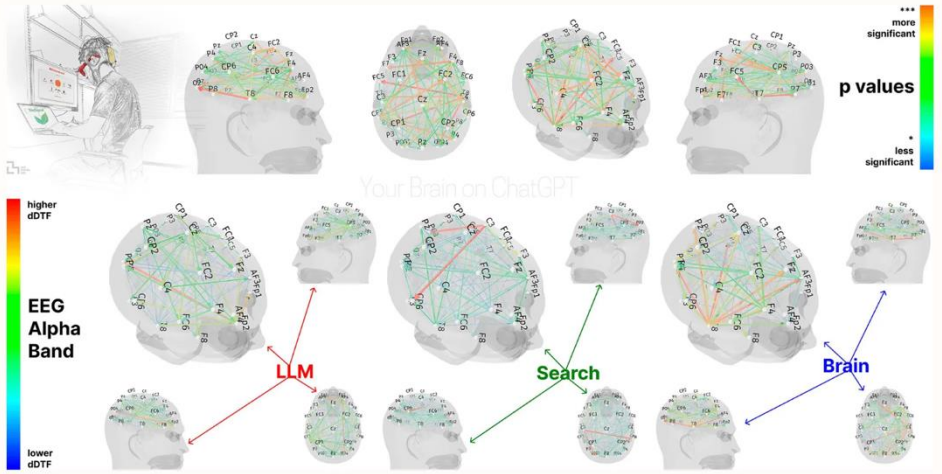
BUSINESS TECH

ChatGPT May Be Eroding Critical Thinking Skills, According to a New MIT Study

ADD TIME ON GOOGLE

by Andrew R. Chow
CORRESPONDENT

Updated: NOV 13, 2025 3:45 PM ET • Published: JUN 17, 2025 3:40 PM ET



A visualization of a new study on AI chatbots by MIT Media Lab scholars. *Nataliya Kosmyna*

The future of Computer Science

AI is unavoidable

AI speeds things up but

slows your brain down

Code review might be the next big thing

The future of Computer Science

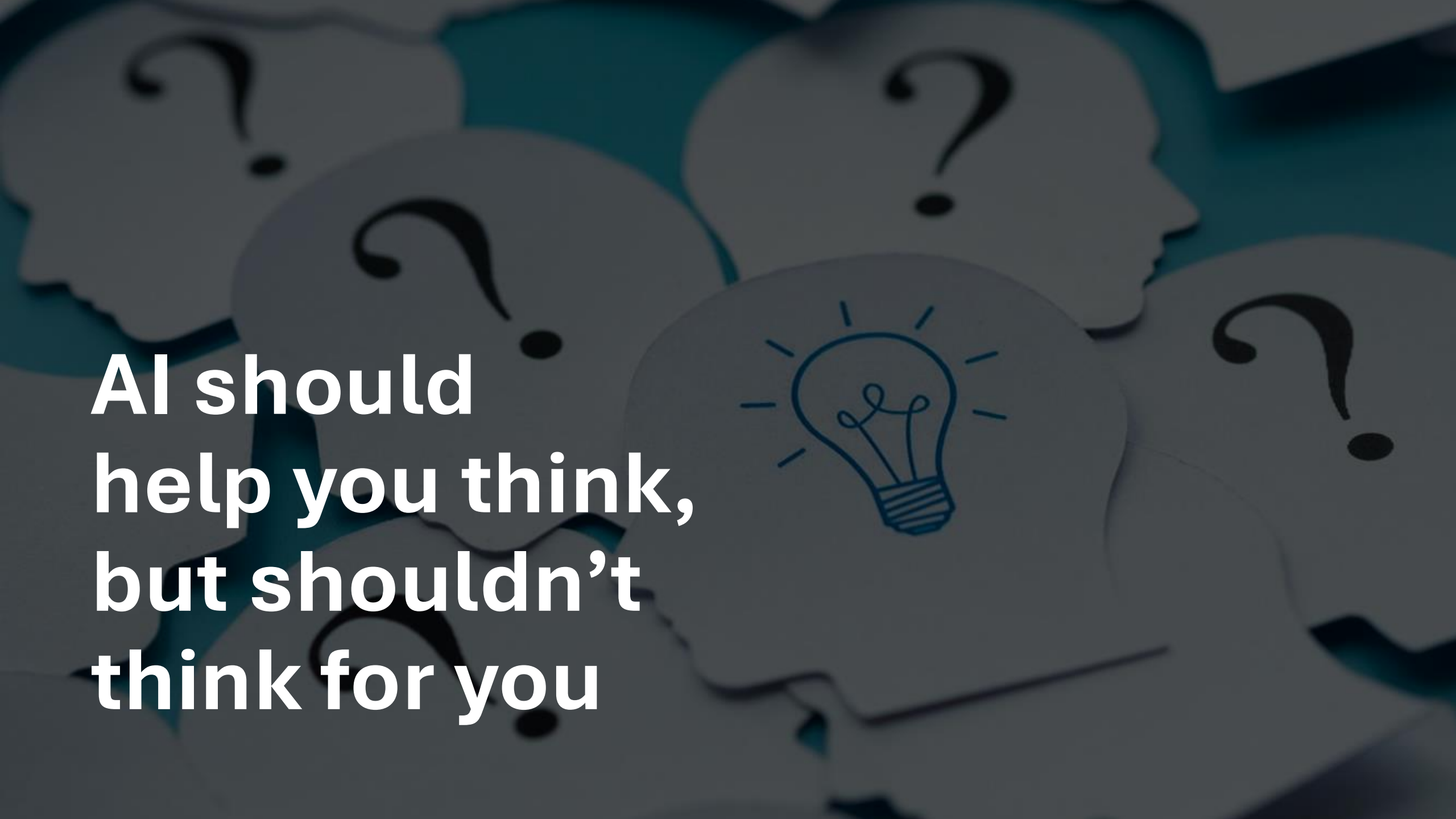
AI speeds things up but

slows your brain down

Code review might be the next big thing

RFL (Rust for Linux) project

- Bottleneck is code review
- Even with LLM


AI should
help you think,
but shouldn’t
think for you

In-Class Activity

- Goal: Evaluate LLMs' performance on new problems from LeetCode
- LeetCode's weekly contest releases 4 "new" problems every week
- We will work on:
  - Weekly Contest 499 (just released 2 days ago) AND
  - Weekly Contest 395 (2 years ago)
- Each group will prompt an LLM of your choice to solve 4 problems from either contest 499 or 395 **(depends on your group assignment)**
- Using either Java or Rust **(depends on your group assignment)**

Scoreboard!!!

- Scoreboard Link

References

1. Kalliamvakou, 2022. https://github.blog/news-insights/research/research-quantifying-github-copilots-impact-on-developer-productivity-and-happiness/
2. Scheffler, 2023. https://www.freethink.com/robots-ai/github-copilot
3. H. Pearce, B. Ahmad, B. Tan, B. Dolan-Gavitt, and R. Karri. Asleep at the keyboard? assessing the security of github copilot's code contributions. In Proc. IEEE Symposium on Security and Privacy (S&P), 2022.
4. A. K. Zhang, N. Perry, R. Dulepet, E. Jones, J. W. Lin, J. Ji, C. Menders, G. Hussein, S. Liu, D. Jasper, et al. Cybench: A framework for evaluating cybersecurity capabilities and risk of language models. arXiv
5. Y. Zhang, Y. Xie, S. Li, K. Liu, C. Wang, Z. Jia, X. Huang, J. Song, C. Luo, Z. Zheng, R. Xu, S. Liu, Y.and Zheng, and X. Liao. 2025. Unseen Horizons: Unveiling the Real Capability of LLM Code Generation Beyond the Familiar. In 2025 IEEE/ACM 47th ICSE.
6. F. Liu, Y. Liu, L. Shi, H. Huang, R. Wang, Z. Yang, L. Zhang, Z Li, and Y. Ma. 2024. Exploring and Evaluating Hallucinations in LLM-Powered Code Generation. arXiv
7. S. Li, Y. Cheng, J. Chen, J. Xuan, S. He, and W. Shang. 2024. Assessing the Performance of AI-Generated Code: A Case Study on GitHub Copilot. In 2024 IEEE 35th ISSRE.
8. Y. Chen, M. Chen, C. Gao, Z. Jiang, Z. Li, and Y. Ma. 2025. Towards Mitigating API Hallucination in Code Generated by LLMs with Hierarchical Dependency Aware. In Proceedings of the 33rd ACM International Conference on the Foundations of Software Engineering.
9. C. J. Chong, M. Ahmed, Z. Yao, and I. Neamtiu. 2026. Can LLMs be Effective Code Contributors? A Study on Open-source Projects. EASE.