

Lecture 26: Finale

CS 5150, Spring 2025

Administrative Reminders

- Final Delivery May 14, 12 PM
- Final Presentation by May 10
 - Reserve remaining time for final report/ handover package preparation
 - Not for feature development
- What to present?
 - Milestones: Promised vs Delivered
 - Demo (client may ask to show additional use cases)
 - User Study results
 - Testing status
 - Integration status

In-Class Exam 2 Stats

- Min: 17, Max: 44, Mean: 31, Median: 31
- Grades released on gradescope: In class exam, Report #4 ongoing

Fall 2025 Courses: CS 6158

- CS 6158: Software Engineering in the Era of Machine Learning
- **Instructor:** Saikat Dutta
- **Goals:**
 - Study state-of-the-art research ideas in SExML
 - Hands-on exposure to Software Engineering research
 - Apply machine learning-based techniques to solve software engineering problems
 - Apply automated software engineering techniques to machine learning systems.
 - Develop and implement new research ideas
- Fall 2024 version: <https://www.cs.cornell.edu/courses/cs6158/2024fa>
- Apply for TA!

Fall 2025 Courses: CS 5154

- CS 5154: Software Testing
- **Instructor:** Owolabi Legunsen
- **Goals:**
 - Deep dive into testing: regression testing, unit testing, mutation analysis, ...
 - Design and automate the execution of high-quality software tests.
 - Generate test suites that meet coverage and other adequacy criteria.
- **Project:** Extend your 5150 project to focus on testing
- Apply for TA!

Lecture Goals

- Few notes about Ethics
- Brief overview of AI/ML for SE landscape
- And using SE techniques to solve AI/ML related challenges

Professionalism & Ethics

What should you do if you discover a major security vulnerability in a piece of widely-used software?

Responsible disclosure

- AKA "coordinated vulnerability disclosure"
- Coordinate timing of announcement with vendor
 - Give them time to patch products, prepare press response
 - Upper bound on timing to hasten vendor action (typ. 90 days)
- For open-source projects, look for security policy (SECURITY.md)
 - Contact Vulnerability Management Team or owner
 - Do not post details to public mailing lists, chat rooms
- May be assigned placeholder CVE to coordinate efforts without disclosing details

Which of these development efforts would you be comfortable contributing to?

- Drug marketing campaign
- Click fraud
- Selling 0-day vulnerabilities
- Reverse engineering
- Weaponized AI
- Selling personal data
- Bitcoin mining

Ethics

- Software can harm society beyond physical injury
- Personal fulfilment is important too
 - Take responsibility for your work
 - Avoid future regrets
- Compared to traditional engineering, software has *less oversight* and *wider impact*
 - Amplification: One day's work can affect millions of people, consume millions of hours

Diversity

- Wider impact => more diverse user base
 - => More potential to reinforce stereotypes, inequity
- Failure to anticipate/respond to biased systems can lead to major societal (not to mention reputational) harm
- Need to expand diversity during development (shift left)
 - More diverse developer teams
 - More diverse user testing
- "Single source of truth" does not apply to human society
 - Disputed borders
 - Different interpretations of words/phrases/symbols
 - Different value systems

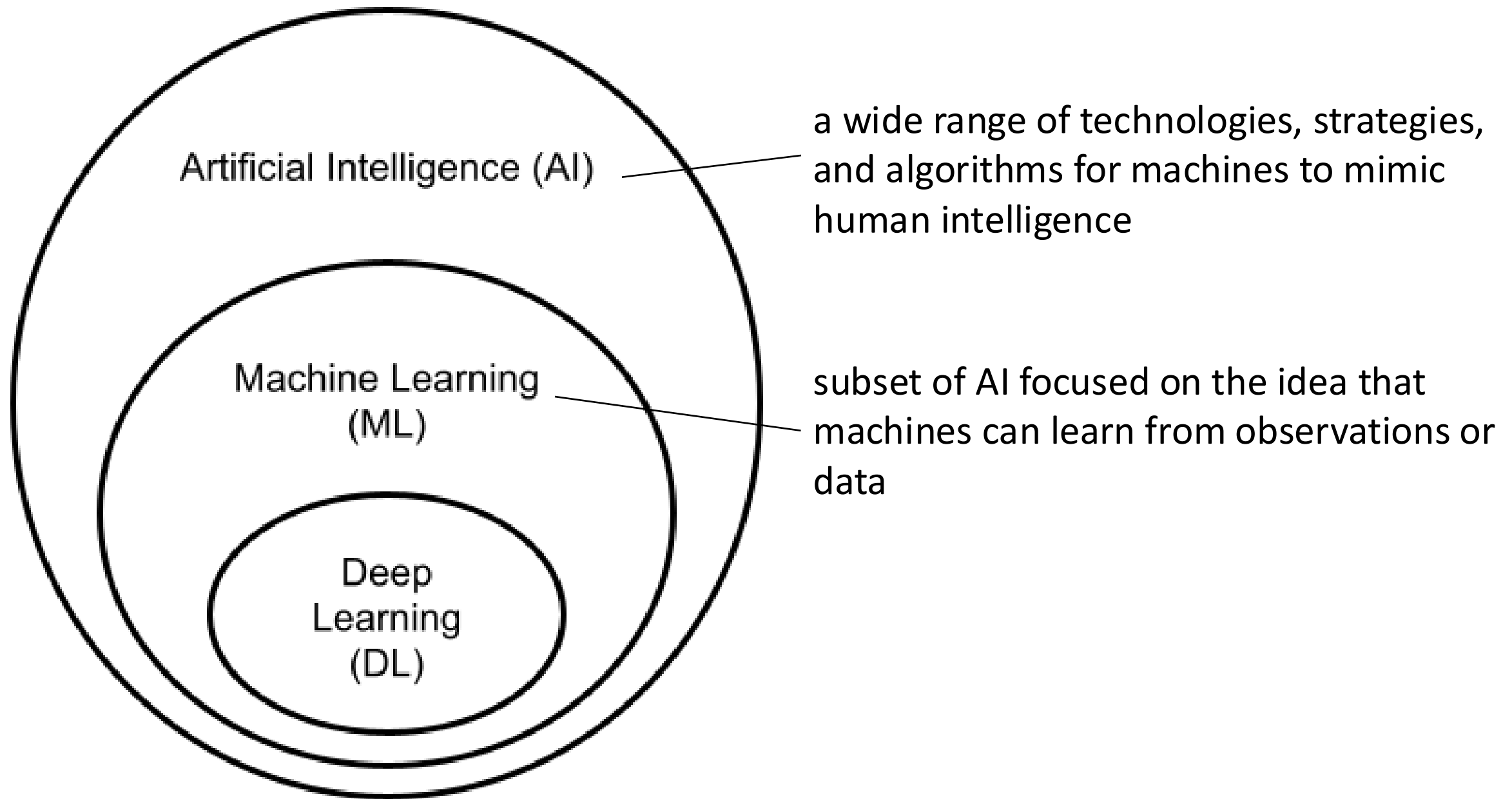
Ethics extends beyond code

- Hiring practices
 - Beware affinity bias, groupthink
- Promotions/opportunities
 - Beyond mentoring - [advocate](#) for coworkers who do good work but seem to go unnoticed
- Decision-making
 - Don't defend decisions solely on precedent
 - Look beyond direct “bottom line” impact

ACM Code of ethics and professional practice

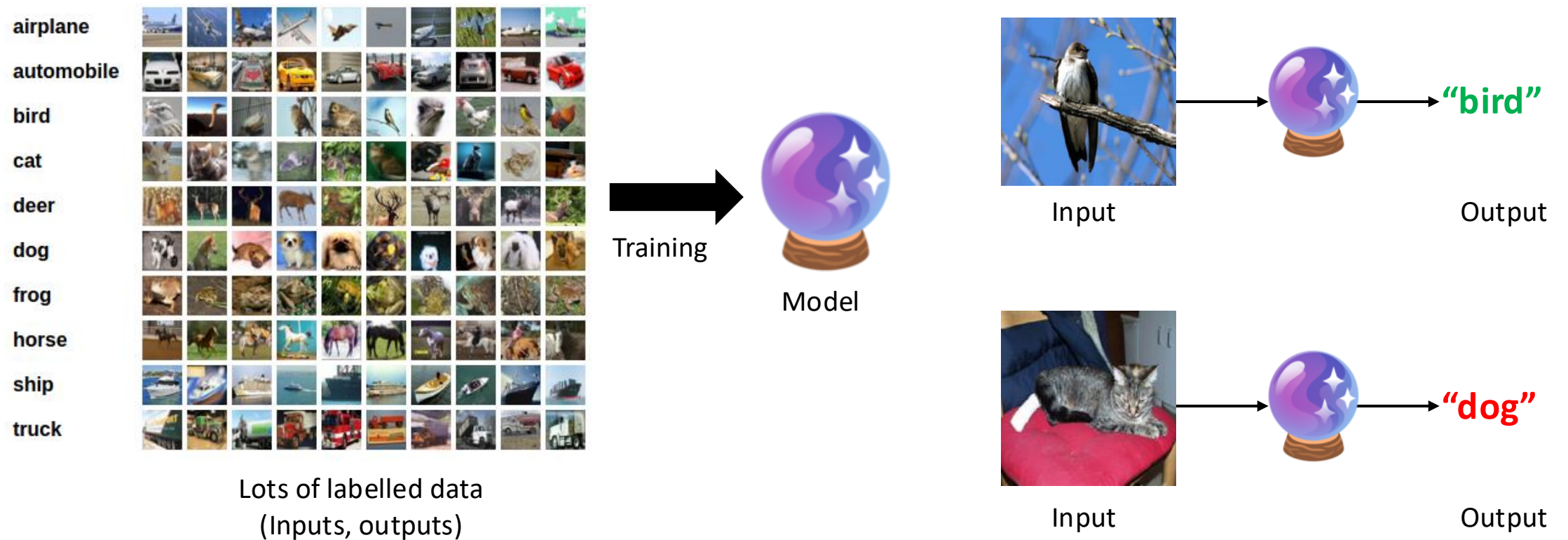
1. PUBLIC – Software engineers shall act consistently with the public interest.
2. CLIENT AND EMPLOYER – Software engineers shall act in a manner that is in the best interests of their client and employer consistent with the public interest.
3. PRODUCT – Software engineers shall ensure that their products and related modifications meet the highest professional standards possible.
4. JUDGMENT – Software engineers shall maintain integrity and independence in their professional judgment.
5. MANAGEMENT – Software engineering managers and leaders shall subscribe to and promote an ethical approach to the management of software development and maintenance.
6. PROFESSION – Software engineers shall advance the integrity and reputation of the profession consistent with the public interest.
7. COLLEAGUES – Software engineers shall be fair to and supportive of their colleagues.
8. SELF – Software engineers shall participate in lifelong learning regarding the practice of their profession and shall promote an ethical approach to the practice of the profession.

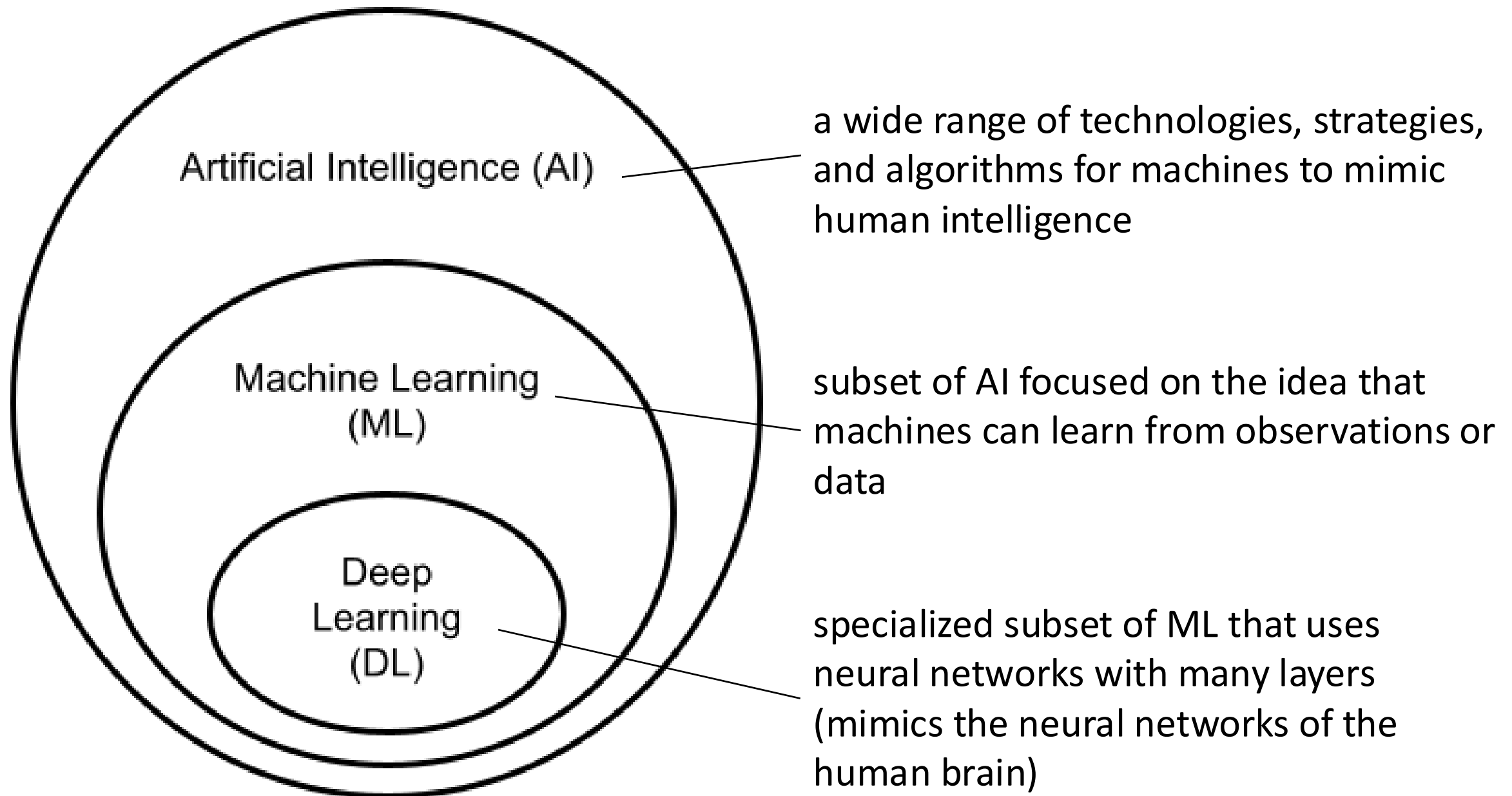
<https://ethics.acm.org/code-of-ethics/software-engineering-code>



Machine Learning in One Slide

(Supervised)



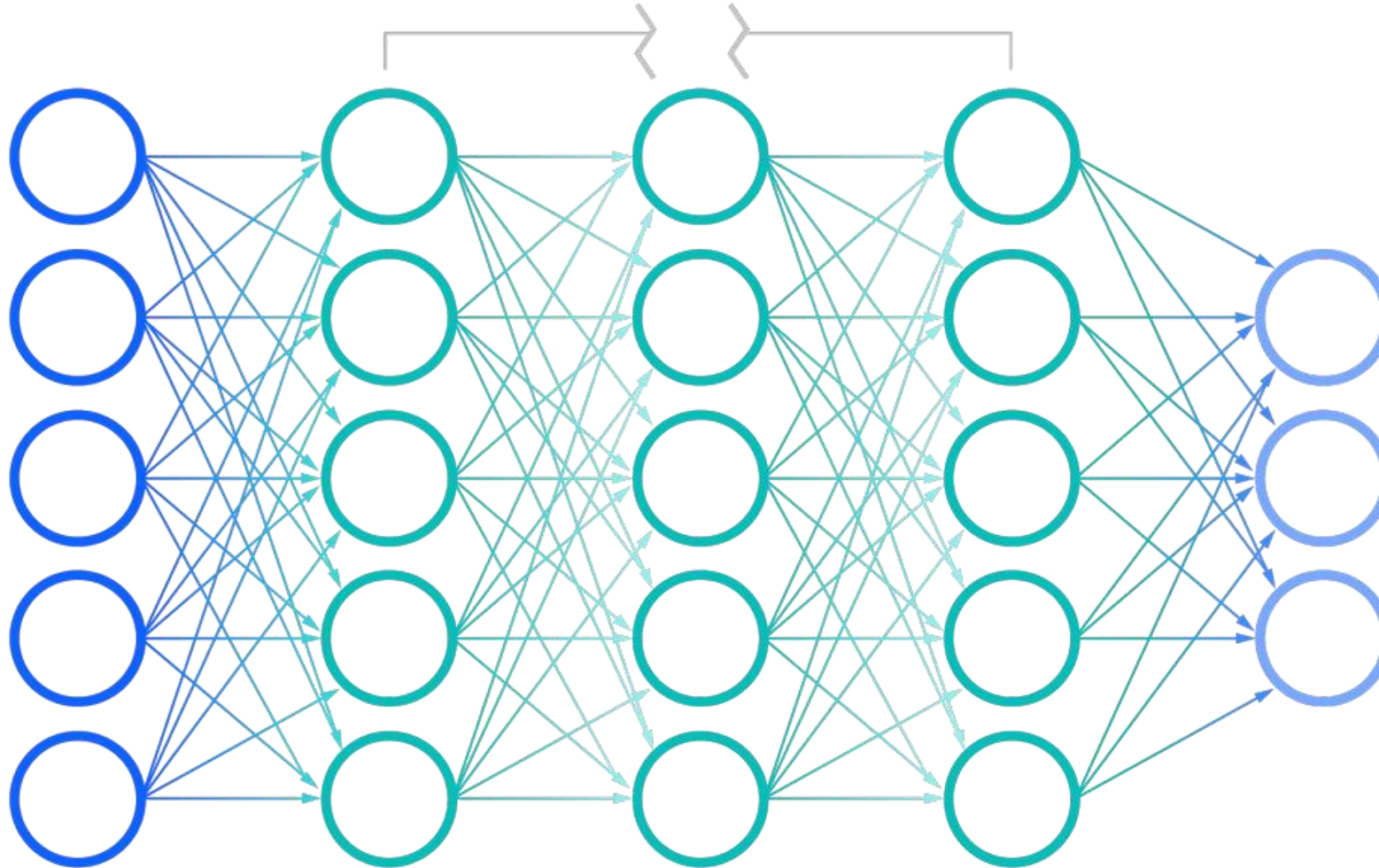


Deep neural network

Input layer

Multiple hidden layers

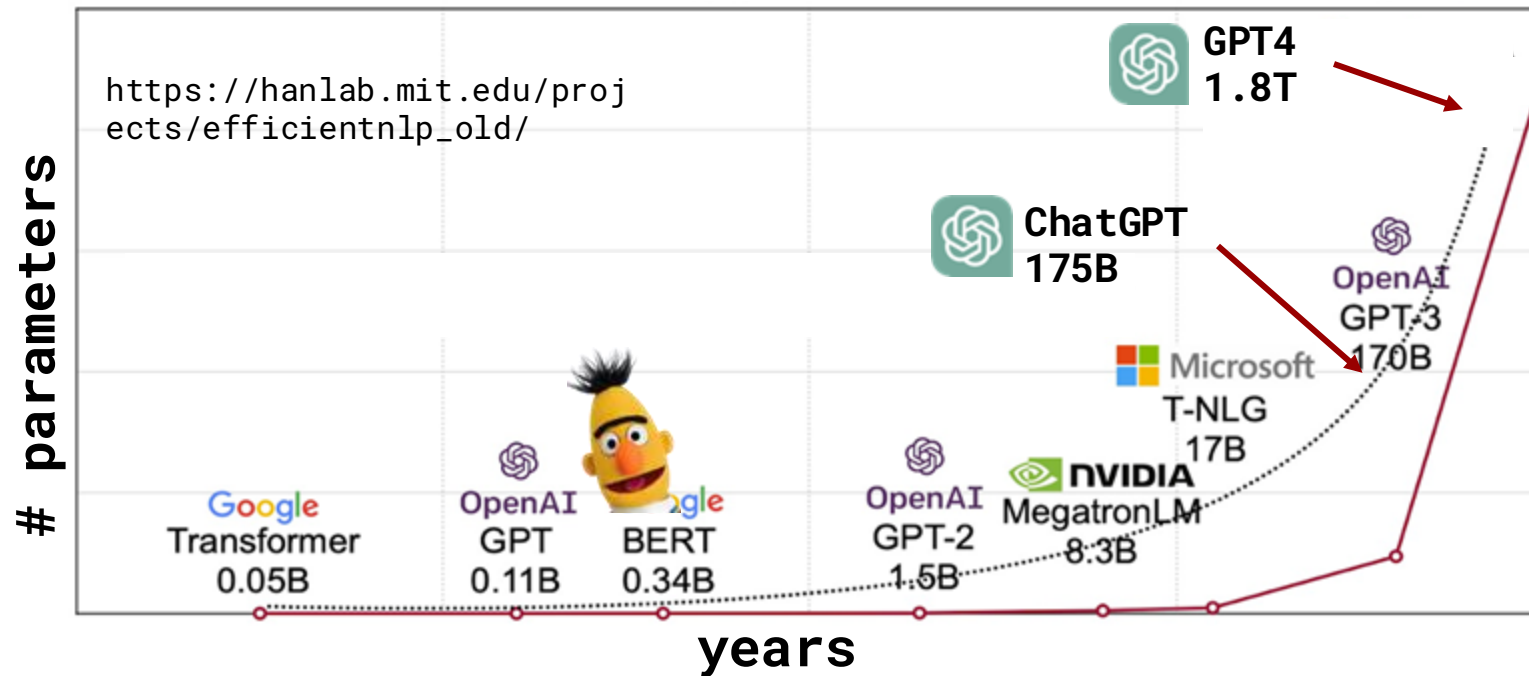
Output layer



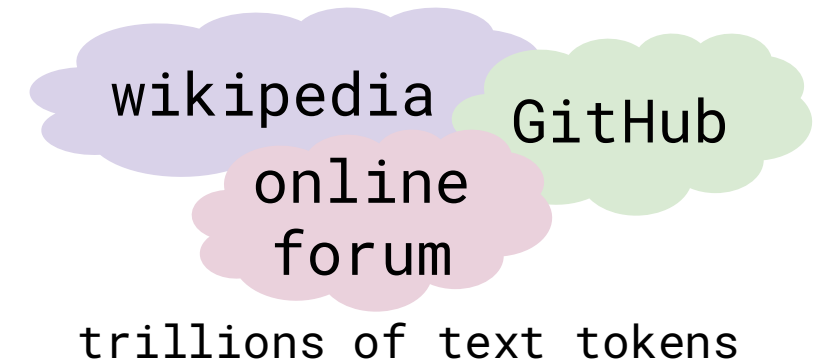
How can ML be useful in SE?

- Automation and reducing manual efforts
 - automate repetitive tasks such as **code generation**, **bug detection**, and **code reviews**
 - AI-powered tools and IDEs for code **autocompletion** and **real-time suggestions**
- Support in problem-solving and decision-making
 - analyze large volumes of data to **uncover patterns** and insights for informed decision-making in project management, etc.
 - process and interpret vast amounts of textual data (documentation, logs, etc.), assisting in efficient diagnostics and troubleshooting

Emergence of Large Language Models (LLMs)



Self-supervised learning on ...



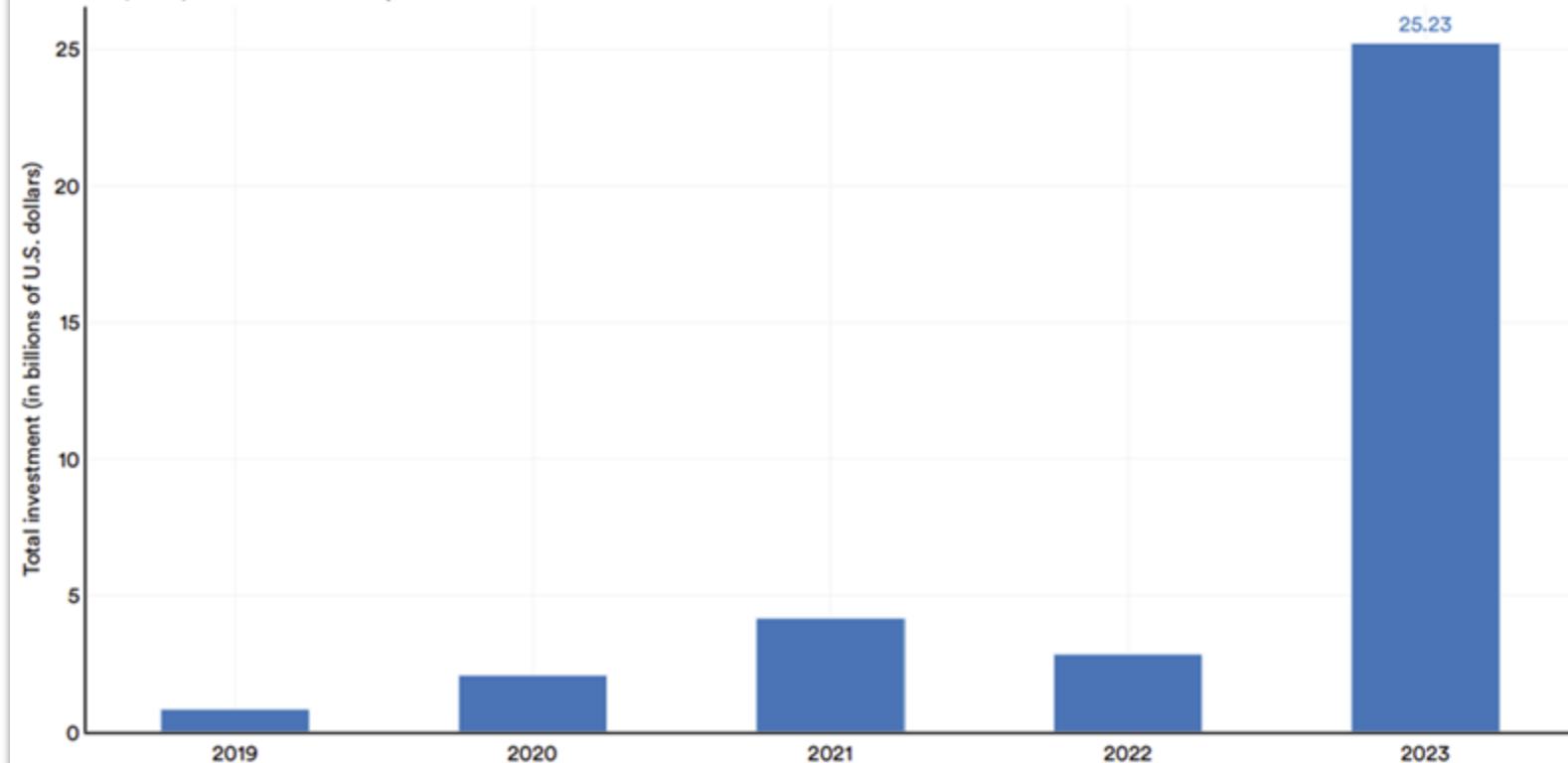
World is
throwing
LLMs at
everything



While overall AI private investment decreased last year, funding for generative AI sharply increased (Figure 4.3.3). In 2023, the sector attracted \$25.2 billion, nearly nine times the investment of 2022 and about 30 times the amount from 2019. Furthermore, generative AI accounted for over a quarter of all AI-related private investment in 2023.

Private investment in generative AI, 2019–23


Source: Quid, 2023 | Chart: 2024 AI Index report



Code Generation and Assistance



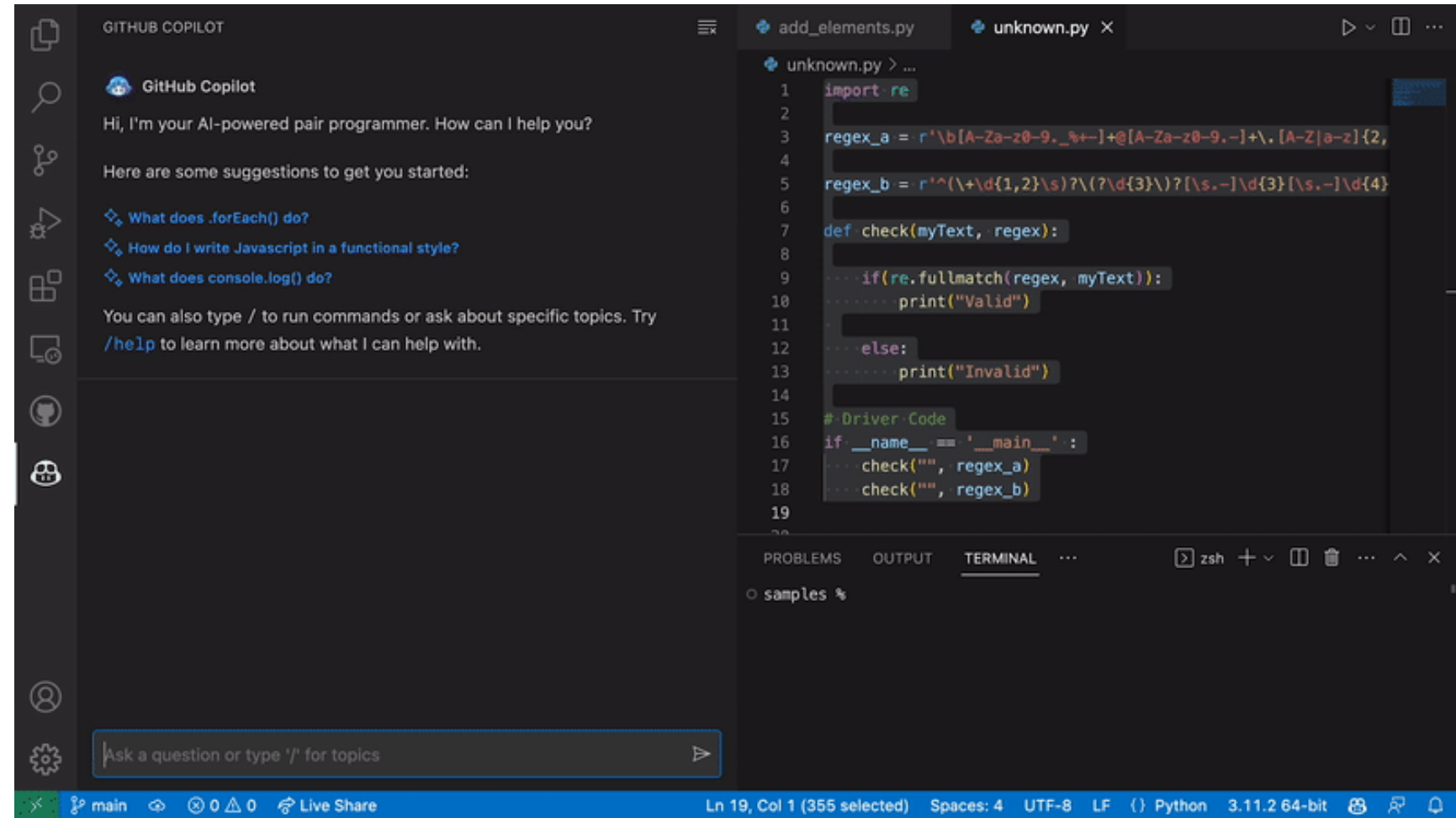
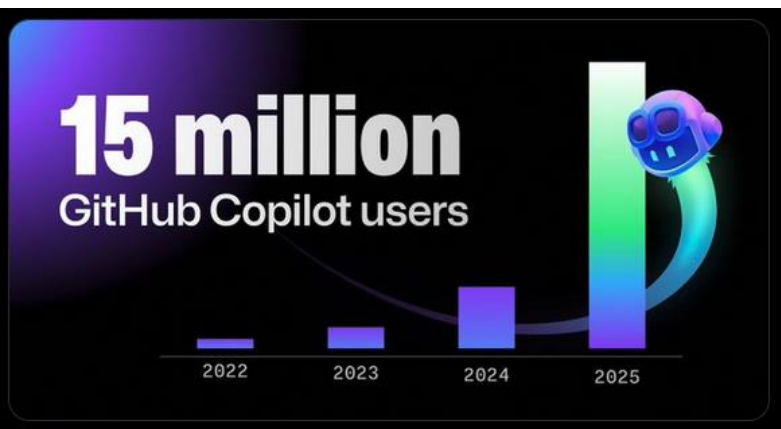
JS test.js 1 ●

JS test.js >  calculateDaysBetweenDates

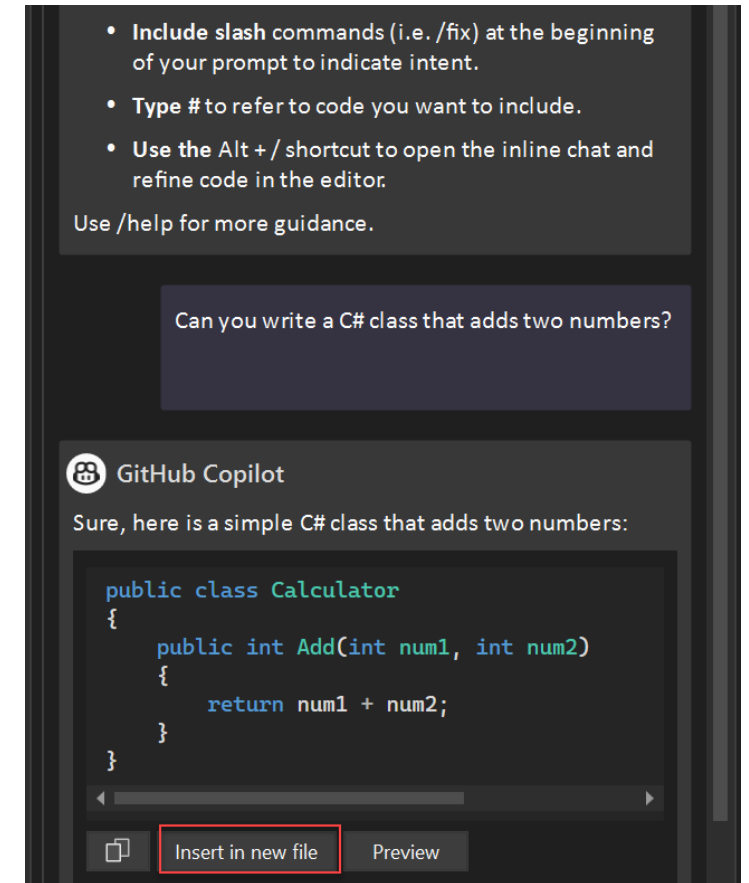
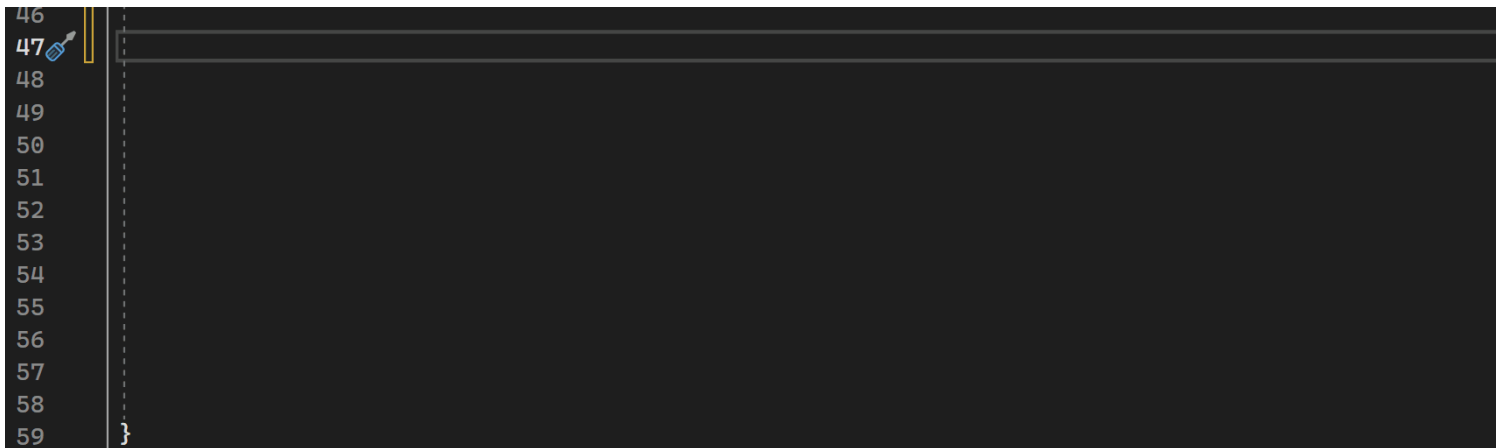
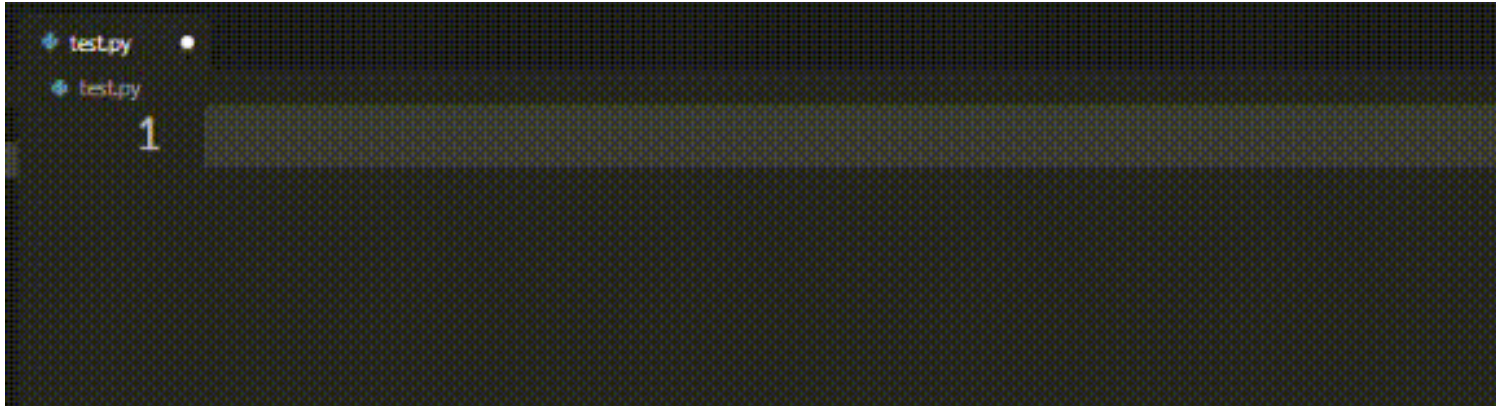
```
1 function calculateDaysBetweenDates(begin, end) {  
    var beginDate = new Date(begin);  
    var endDate = new Date(end);  
    var days = Math.round((endDate - beginDate) / (1000 * 60 * 60 * 24));  
    return days;  
}  
2
```

Github Copilot

- Code Completion
- Code Analysis
- Fixing issues



Generate Code in Different Ways



Automated Code Reviews



SonarQube interface showing project issues for eShopOnWeb.

Navigation: sonarqube Projects Issues Rules Quality Profiles Quality Gates Administration

Search: Search for projects... A

Project: eShopOnWeb main

Date: April 23, 2024 at 11:53 PM Version not provided

Overview Issues Security Hotspots Measures Code Activity

Project Settings Project Information

Type: CODE SMELL Clear

Bug 20

Vulnerability 0

Code Smell 151

Press ⌘ to add to selection

Severity

Blocker	0	Minor	28
Critical	5	Info	62
Major	56		

Scope

Resolution

Bulk Change

1 / 151 issues 1d 6h effort

src/ApplicationCore/Constants/AuthorizationConstants.cs

- Add a 'protected' constructor or the 'static' keyword to the class declaration. 4 years ago L3 design
- Complete the task associated to this 'TODO' comment. 3 years ago L7 cwe
- Complete the task associated to this 'TODO' comment. 3 years ago L10 cwe

src/.../Entities/BuyerAggregate/Buyer.cs

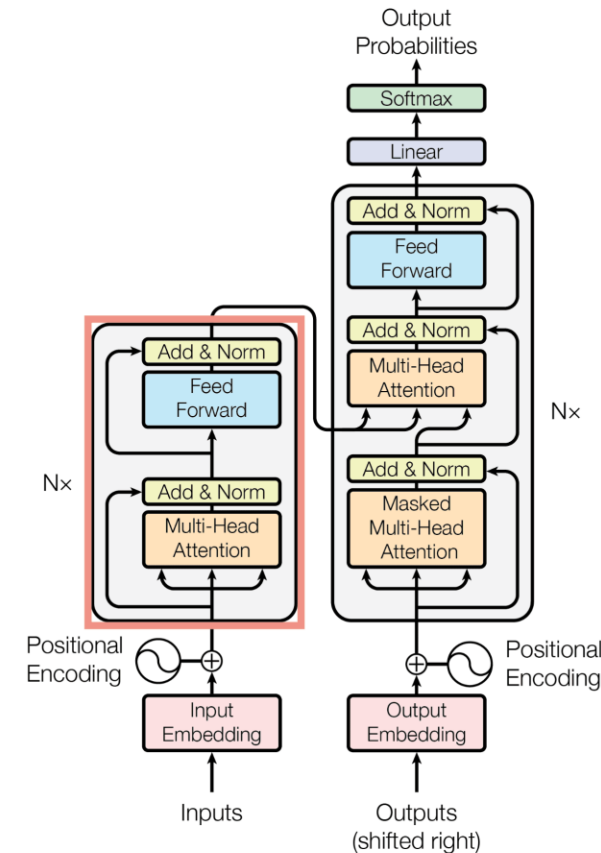
- Make '_paymentMethods' 'readonly'. 6 years ago L11 confusing

Also useful for...

- Writing Tests
- Refactoring Code
- Understanding Code
- Finding security vulnerabilities
- ...

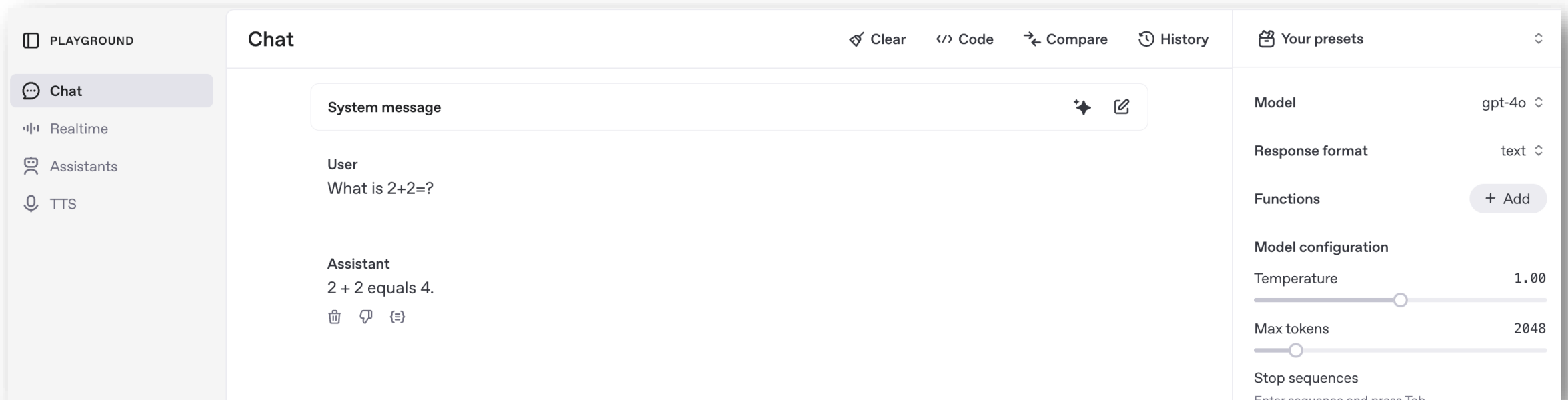
Large Language Models (LLMs)

- Language Modeling: Measure probability of a sequence of words
 - Input: Text sequence
 - Output: Most likely next word
- LLMs are... large
 - GPT-3 has 175B parameters
 - GPT-4 is estimated to have ~1.24 Trillion
- Pre-trained with up to a PB of Internet text data
 - Massive financial and environmental cost



Prompt Engineering

The process of crafting and refining prompts to effectively interact with LLMs to get accurate, relevant, and useful responses.



AI Prompt Engineers Earn \$300k Salaries: Here's How To Learn The Skill For Free

Jodie Cook Contributor ⓘ

I explore concepts in entrepreneurship, AI and lifestyle design.

Follow

Which of these problems should be solved by an LLM? Why or why not?

- Type checking Java code
- Grading mathematical proofs
- Answering emergency medical questions
- Unit test generation for your projects

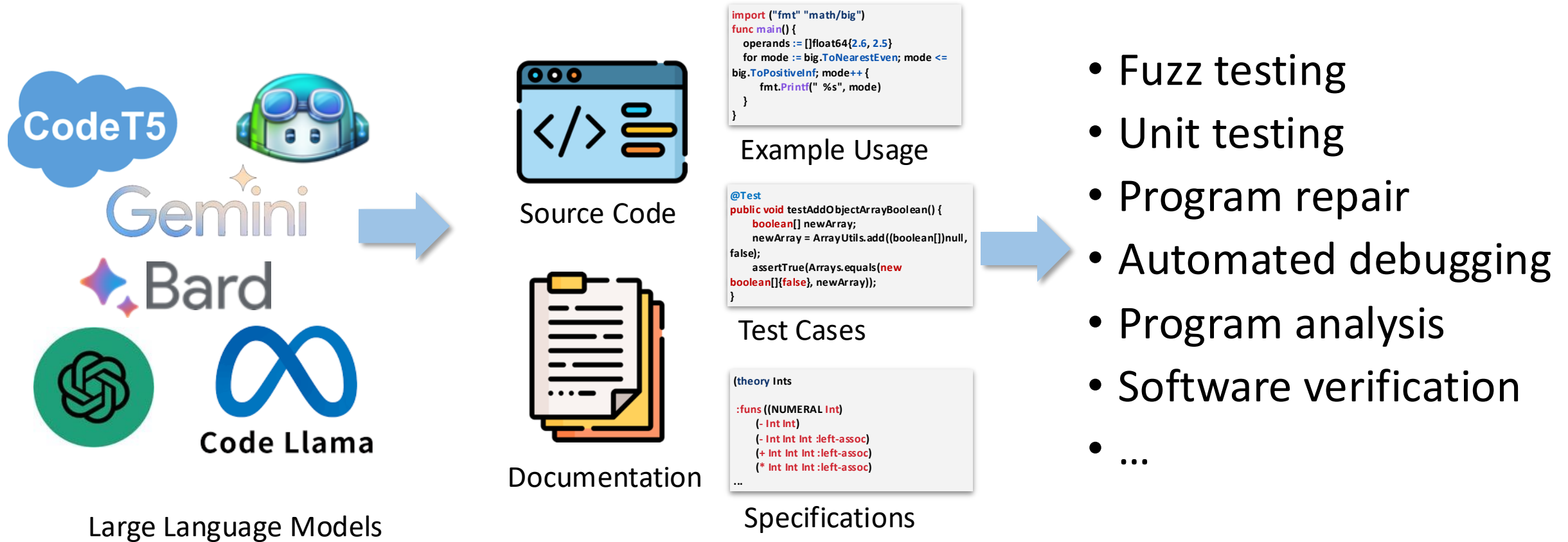
More practical factors to consider when productionizing

- Operational Costs
- Latency/speed
- Intellectual property
- Security

Problems with LLMs

- Hallucinations
 - No guarantees whatsoever
- Limited by prompt length (now upto 100k tokens)
 - Hard to analyze large repos
- Generate Insecure/Inefficient Code (Safety)
- Hard to use for Low-Resource Languages (e.g., Ocaml, Rust, ...)
 - May regurgitate from memory

ML for Quality Assurance (My Research)



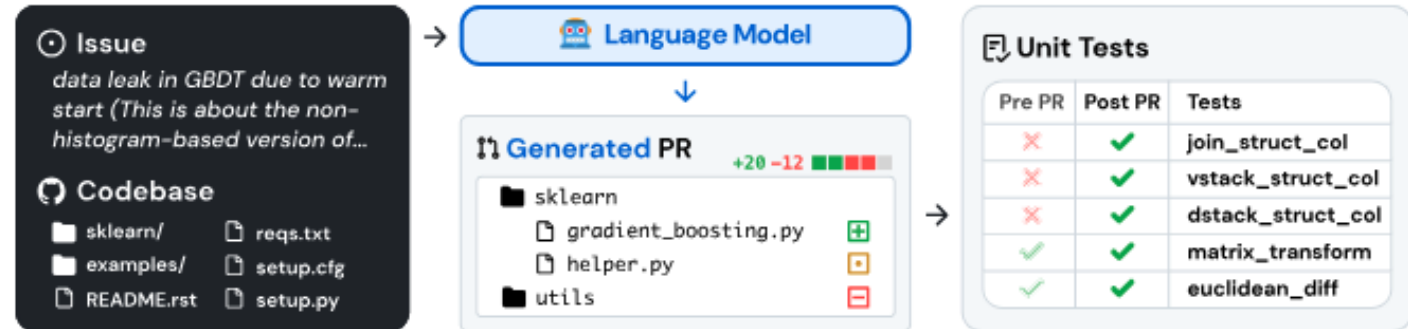
SWE-BENCH: CAN LANGUAGE MODELS RESOLVE REAL-WORLD GITHUB ISSUES?

SWE-Bench (ICLR 24)

- 2024: ~3% (4% on Lite)
- Now: ~33% (65% on Lite)

Carlos E. Jimenez^{*1,2} John Yang^{*1,2} Alexander Wettig^{1,2}
Shunyu Yao^{1,2} Kexin Pei³ Ofir Press^{1,2} Karthik Narasimhan^{1,2}

¹Princeton University ²Princeton Language and Intelligence ³University of Chicago



Leaderboard

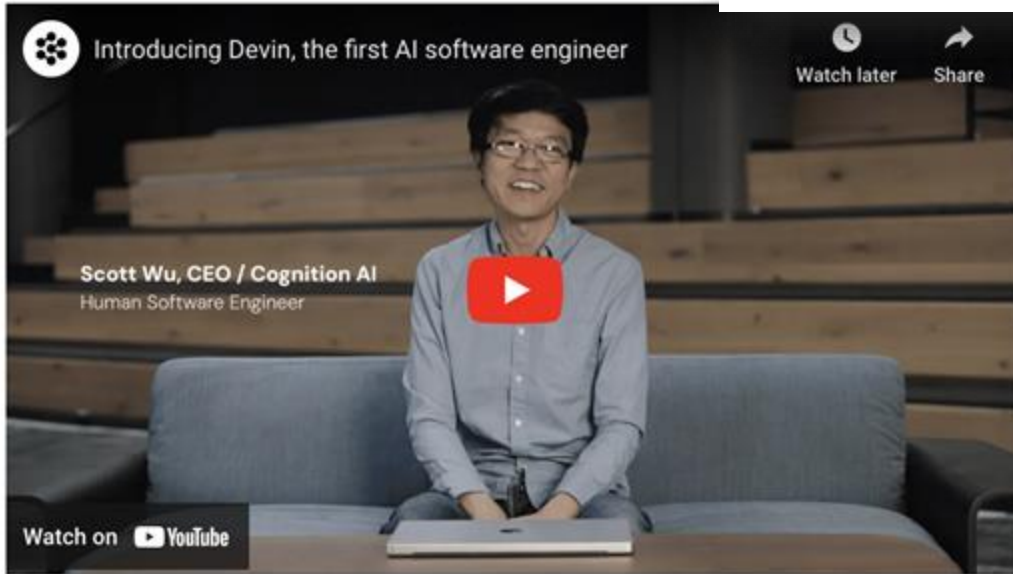
Lite	Verified	Full	Multimodal	
Model				
				% Resolved Org Date
NEW Augment Agent v0				65.40 2025-03-16
W&B Programmer O1 crosscheck5				64.60 2025-01-17
AgentScope				63.40 - 2025-02-06
NEW Tools + Claude 3.7 Sonnet (2025-02-24)				63.20 2025-02-24
NEW EPAM AI/Run Developer Agent v20250219 + Anthropic Claude 3.5 Sonnet				62.80 2025-02-28
CodeStory Midwit Agent + swe-search				62.20 - 2024-12-21
OpenHands + 4x Scaled (2024-02-03)				60.80 2025-02-03
Learn-by-interact				60.20 2025-01-10
devlo				58.20 2024-12-13

BLAME DEVIN | JAN 24, 11:19 AM EST by VICTOR TANGERMANN

The "First AI Software Engineer" Is Bungling the Vast Majority of Tasks It's Asked to Do

It took longer than a human, and failed at the vast majority of tasks.

We've raised a \$21 million Series-A led by Founders Fund. Let's

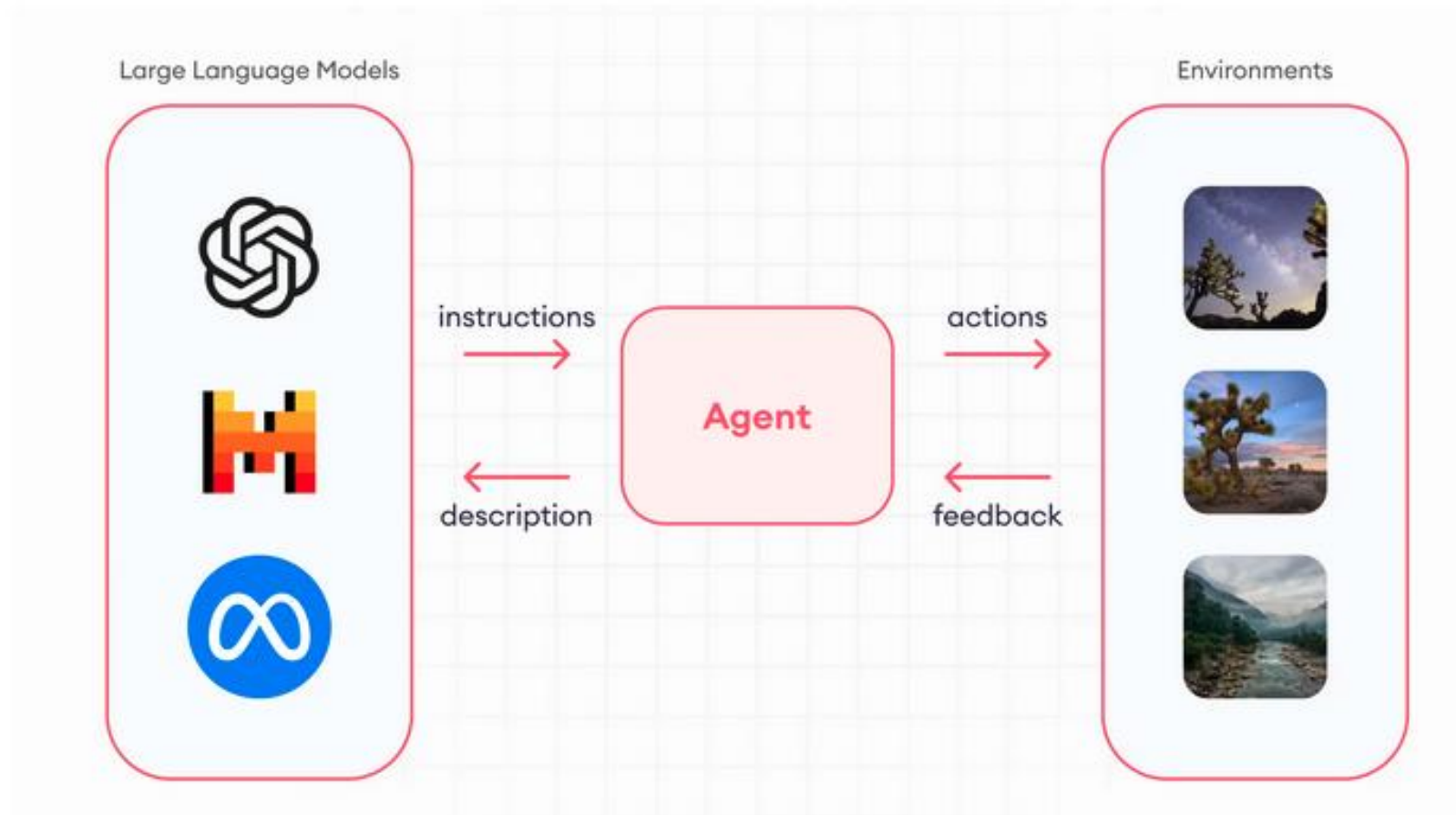


March 12th, 2024 | Written by Scott Wu

Introducing Devin, the first AI software engineer

And setting a new state of the art on the SWE-bench coding benchmark

LLM Agents



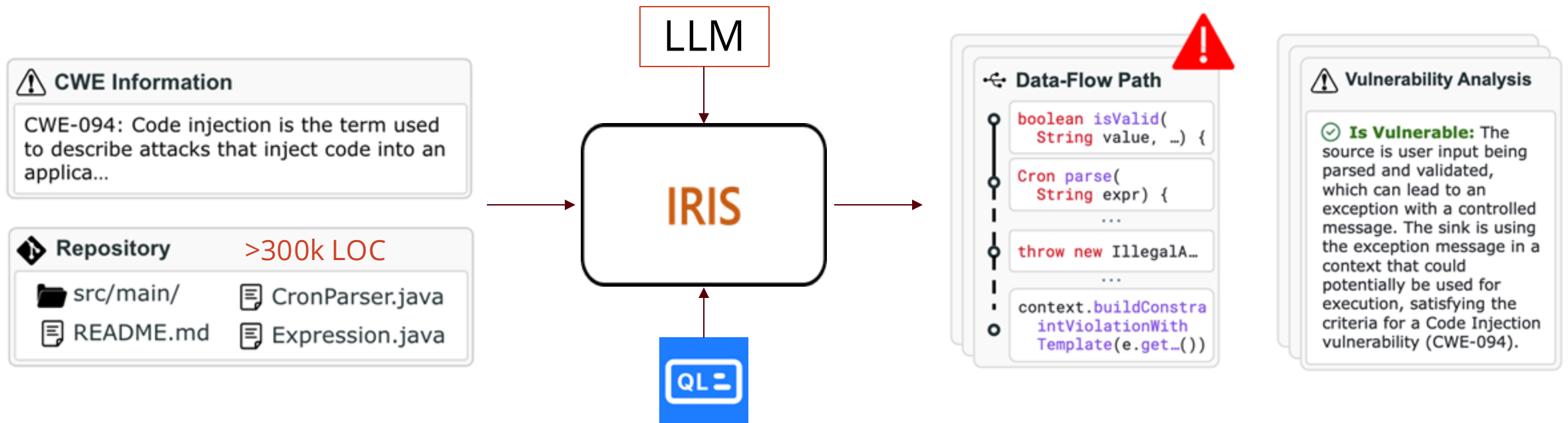
Coding Agents (Demo)



<https://www.anthropic.com/claude-code>

IRIS: Neuro-Symbolic Static Analysis

Combine LLMs with Static Analysis (CodeQL) for whole-repository analysis.



IRIS: LLM-Assisted Static Analysis for Detecting Security Vulnerabilities.

Ziyang Li, Saikat Dutta, Mayur Naik. **ICLR 2025.**

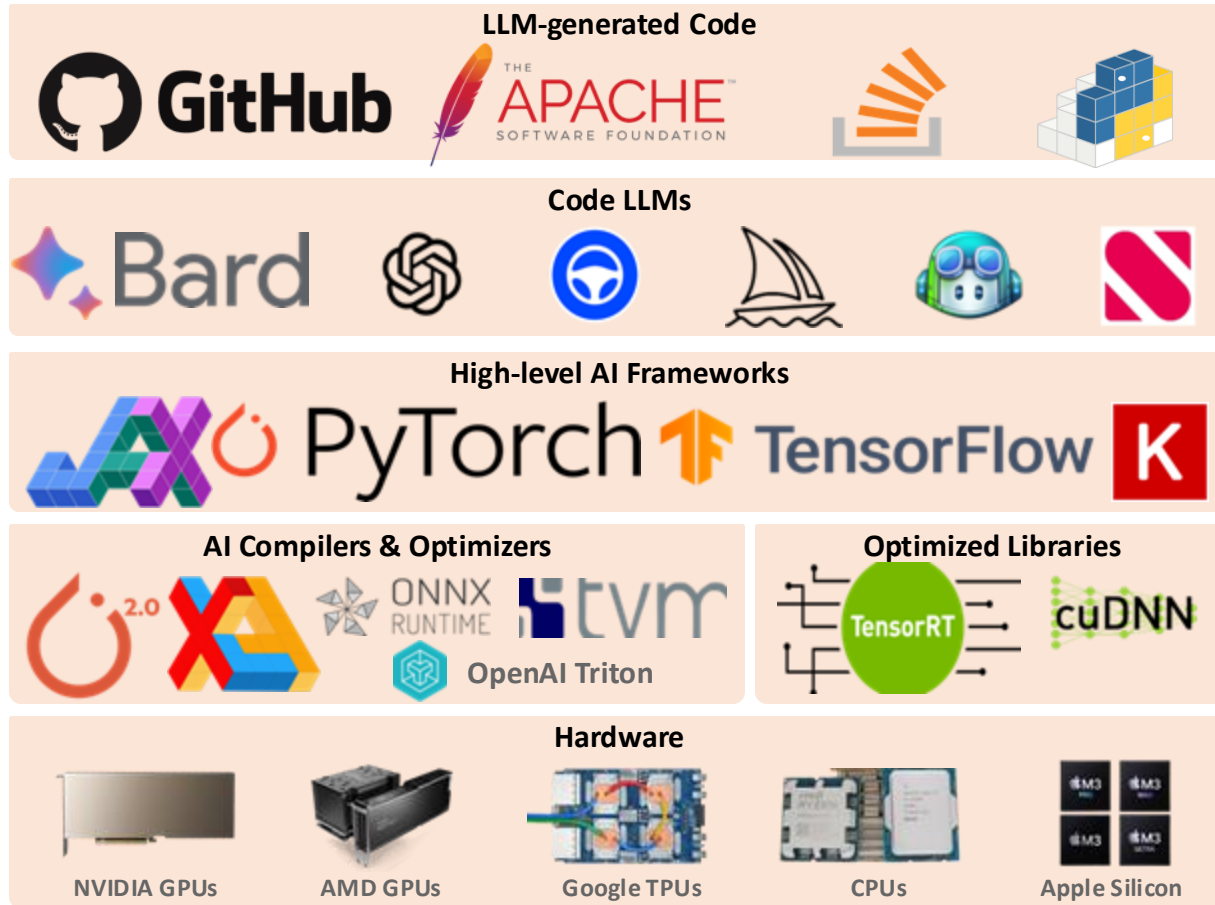
<https://github.com/iris-sast/iris>

IRIS: Main Ideas and Results

- LLMs can be used to **infer the specifications** (source/sink/sanitizers) of commonly used external library APIs
- LLMs can use natural language information to understand code context; can **filter false positives!**

Method		#Detected (/120)	Detection Rate (%)	Avg FDR (%)	Avg F1 Score
CodeQL		27	22.50	90.03	0.076
IRIS +	GPT-4	55 (↑ 28)	45.83 (↑ 23.33)	84.82 (↓ 5.21)	0.177 (↑ 0.101)
	GPT-3.5	47 (↑ 20)	39.17 (↑ 16.67)	90.42 (↑ 0.39)	0.096 (↑ 0.020)
	L3 8B	41 (↑ 14)	34.17 (↑ 11.67)	95.55 (↑ 5.52)	0.058 (↓ 0.018)
	L3 70B	54 (↑ 27)	45.00 (↑ 22.50)	90.96 (↑ 0.93)	0.113 (↑ 0.037)
	DSC 7B	52 (↑ 25)	43.33 (↑ 20.83)	95.40 (↑ 5.37)	0.062 (↓ 0.014)

Quality Assurance for ML



- Benchmarking
- Code correctness
- Code security
- Model security
- System reliability
- ...

Flaky Tests Empirical Study*

First study of flaky tests in **Machine Learning** libraries

Studied 75 flaky tests in 20 ML libraries

- **60%** caused due to **Algorithmic Randomness**: e.g., Sampling, Dropout (using **random number generators**)
- **Adjusting assertion bounds** is the most common fix

How can we automatically **fix** such flaky tests? [**FLEX, FSE'21**]

*Detecting Flaky Tests in Probabilistic and Machine Learning Applications. Saikat Dutta, August Shi, Rutvik Choudhary, Zhekun Zhang, Aryaman Jain, and Sasa Misailovic (ISSTA 2020)

Example Test

```
def test_MCMC_Sampler():
```

```
    sampler = initMCMCSampler(chains=3)
    train_ds = createGaussMixDS()
    result = fit(sampler, train_ds, iters=1000)
```

```
    rvs1 = normal(loc=-1, scale=0.7, n=5000)
    rvs2 = normal(loc=100, scale=0.8, n=5001)
    statistic = ks(result.samples, [rvs1, rvs2])
```

```
    assert statistic < 0.1
```

pyPESTO: provides state-of-art algorithms for optimization and uncertainty analysis of black-box objective functions

Example Test

```
def test_MCMC_Sampler():
```

```
    sampler = initMCMCSampler(chains=3)
    train_ds = createGaussMixDS()
    result = fit(sampler, train_ds, iters=1000)
```

```
    rvs1 = normal(loc=-1, scale=0.7, n=5000)
    rvs2 = normal(loc=100, scale=0.8, n=5001)
    statistic = ks(result.samples, [rvs1, rvs2])
```

```
    assert statistic < 0.1
```

Test is Flaky!

0.01



0.15



0.11



Sources of Randomness

```
def test_MCMC_Sampler():
```

```
    sampler = initMCMCSampler(chains=3)
    train_ds = createGaussMixDS()
    result = fit(sampler, train_ds, iters=1000)
```

```
    rvs1 = normal(loc=-1, scale=0.7, n=5000)
    rvs2 = normal(loc=100, scale=0.8, n=5001)
    statistic = ks(result.samples, [rvs1, rvs2])
```

```
    assert statistic < 0.1
```

Test is Flaky!

0.01



0.15



0.11



Sources of Randomness (MCMC)

```
def propose_parameter(self, x):  
    ...  
    x_new = multivariate_normal.sample(x, cov)  
    return x_new
```

Propose new parameter value

```
def perform_step(self, x, ...):  
    ...  
    u = np.random.uniform(0, 1)  
    if np.log(u) < log_p_acc:  
        x = x_new
```

Accept or Reject new sample

Fixing Flaky Tests in ML Libraries*(FSE'21)

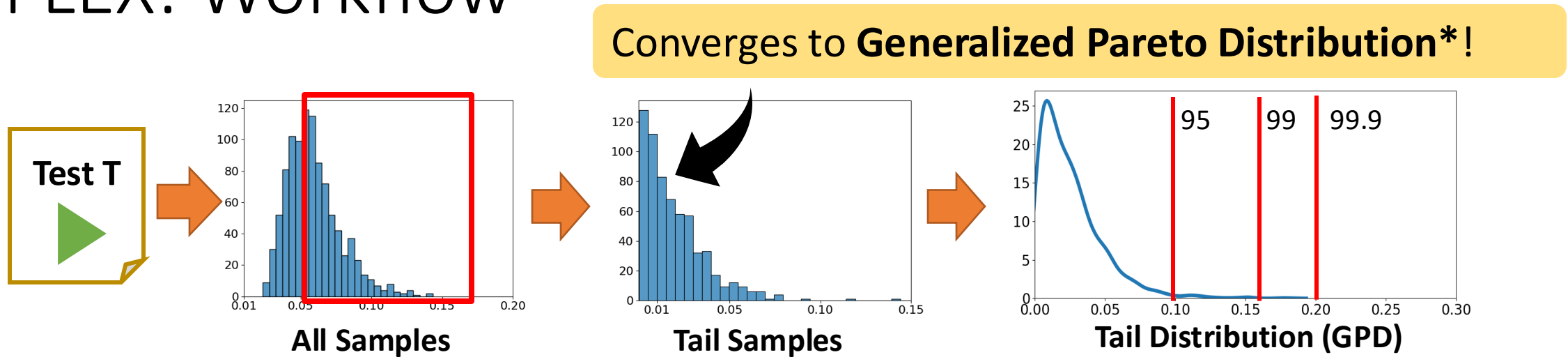
Mitigates **Flakiness** due to **Randomness** of **ML Algorithm**



Statistical Modeling to reason about underlying randomness

* *FLEX: Fixing Flaky Tests in Machine-Learning Projects by Updating Assertion Bounds*. Saikat Dutta, August Shi, and Sasa Misailovic (FSE 2021)

FLEX: Workflow



Challenges for applying Extreme Value Theory:

How to collect I.I.D. samples?



Samples from **Different Test Executions, Same Distribution** are **I.I.D.**

How many samples to collect?

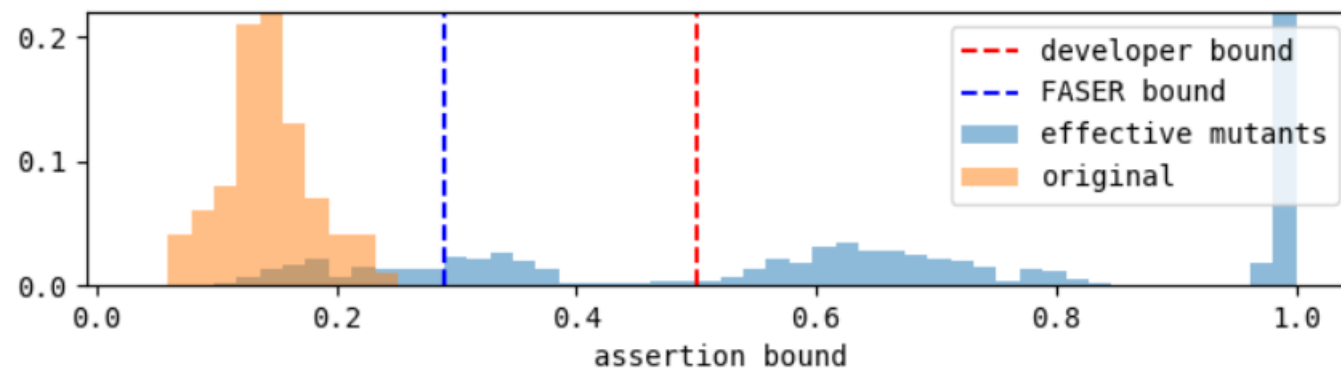


Check convergence using **GPD Convergence Test**

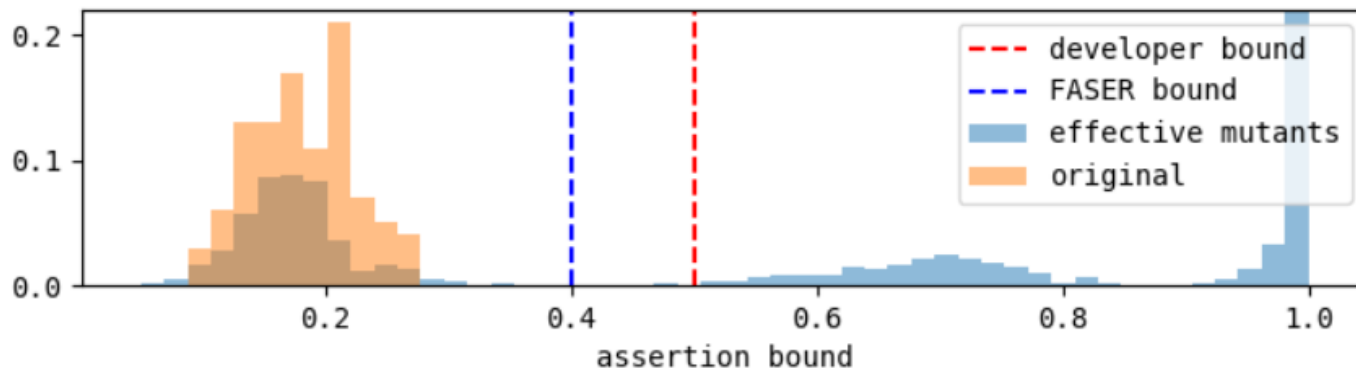
*Pickands et al. – Statistical inference using extreme order statistics (1975)

Effectiveness vs Flakiness*(ICSE 2023)

- **Problem:** Loose Assertion Bound => Flakiness , Effectiveness 
- Balance **Effectiveness** (Mutation Testing) & **Flakiness** (Concentration Inequalities)



Effectiveness **can** be improved!



Effectiveness **cannot** be improved!

- Software engineering is bigger than programming
 - More stakeholders
 - Collaborative development
 - Quality has a cost
- Successful projects involve tradeoffs, communication
 - Different projects warrant different approaches
- Big projects *are* possible
 - With planning & teamwork, can accomplish far more than solo

Good luck with all your future endeavors!