Introduction

CS 6501: AI Systems

Fall 2026

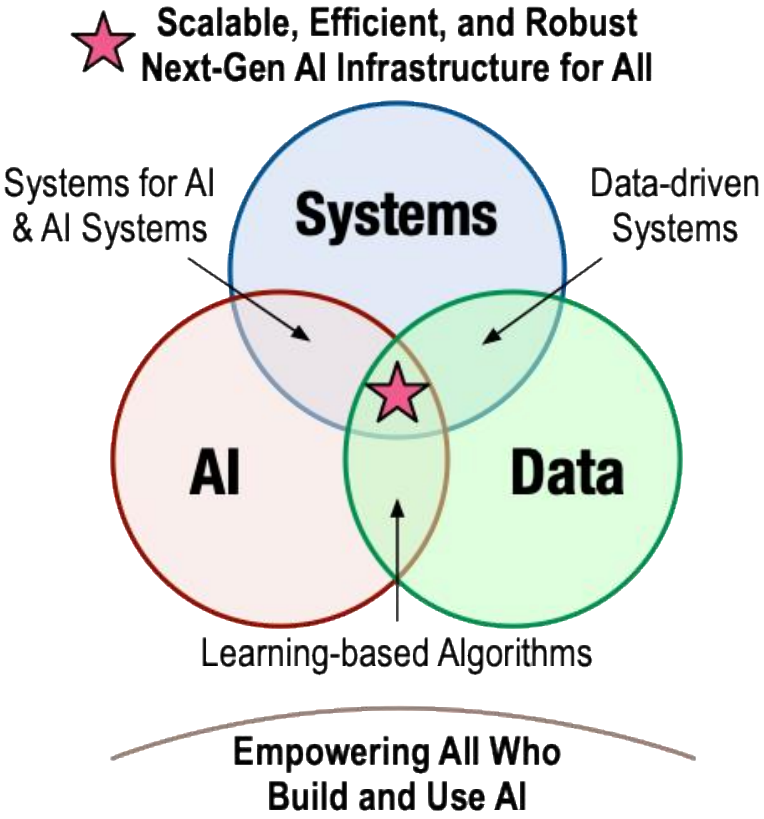
Lecture 1

Yue Cheng

Introduction

- Yue Cheng
  - Associate professor of Data Science & Computer Science
  - Web: https://tddg.github.io
  - Email: mrz7dp@virginia.edu

- Current research: Building better AI systems for all



Course staff and office hours

- Instructor: **Yue Cheng**
  - Office hours: M/W, 2:30pm—3:30pm on Zoom

- GTA: **Saleha Muzammil**
  - Office hours: M/W, 1pm—2:30pm
  - Rice Hall Rm 442

Discussion and getting help

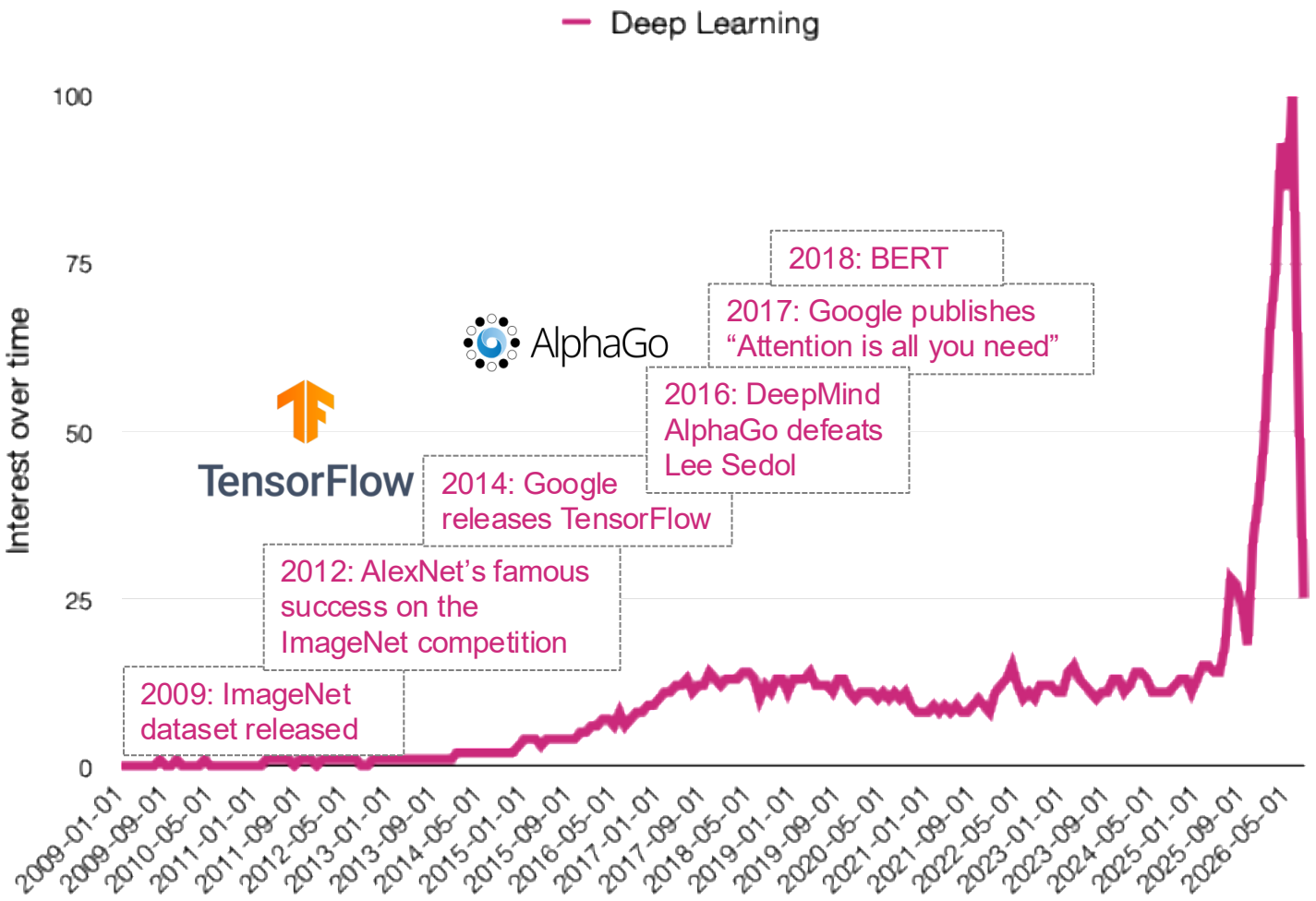
- Discussion, questions: Ed
  - https://edstem.org/us/dashboard
  - Alternative place to ask questions, brainstorming, and discussing anything related to course topics
  - No anonymous posts or questions
  - Can use private posts to instructor/GTA
  - We are monitoring Ed several times a day
  - We will respond to questions in a batch manner

Today's agenda

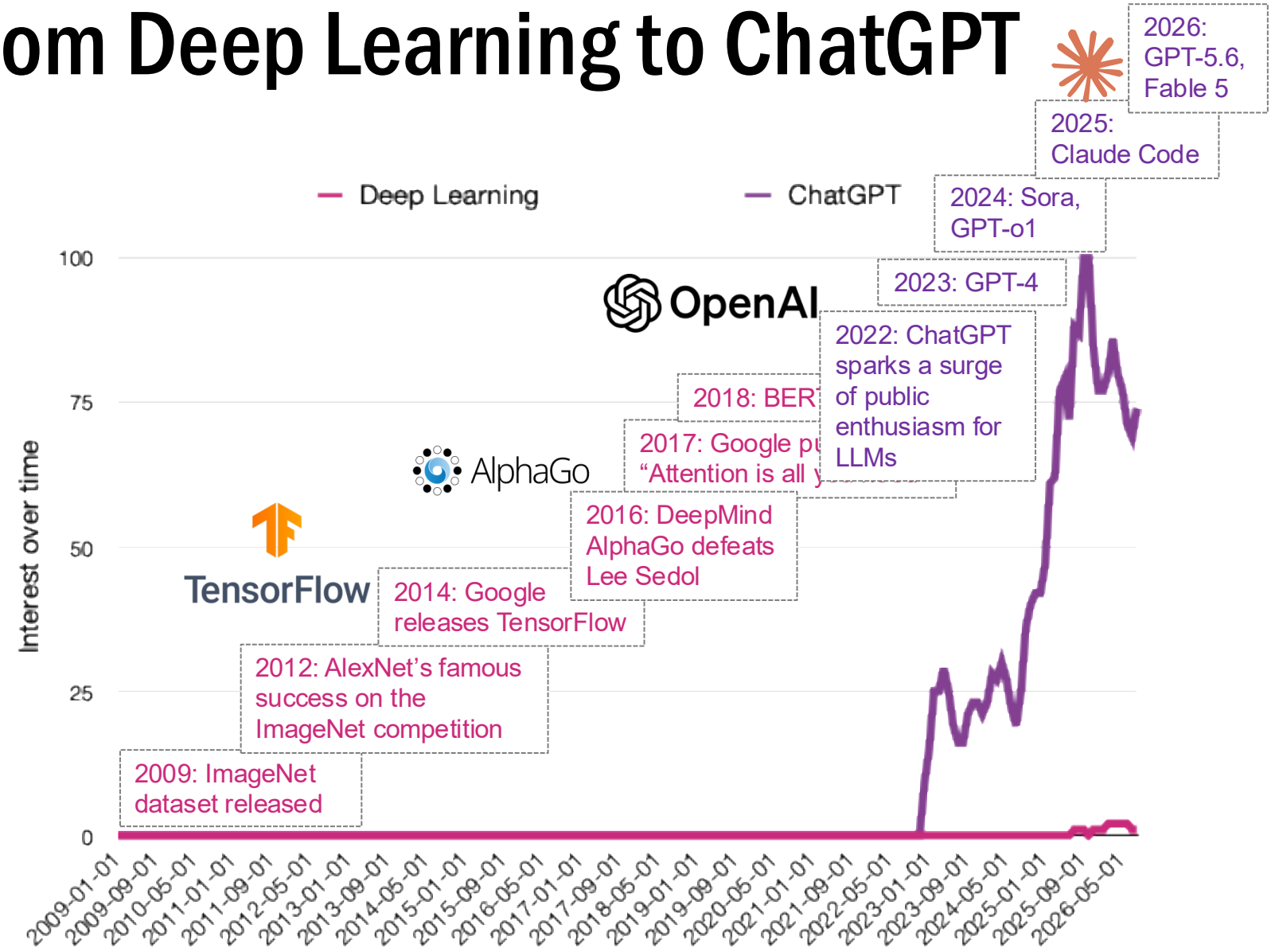
- Why are we studying AI systems?

- What will you do in this course?

Modern AI thru the lens of Google Trends



From Deep Learning to ChatGPT

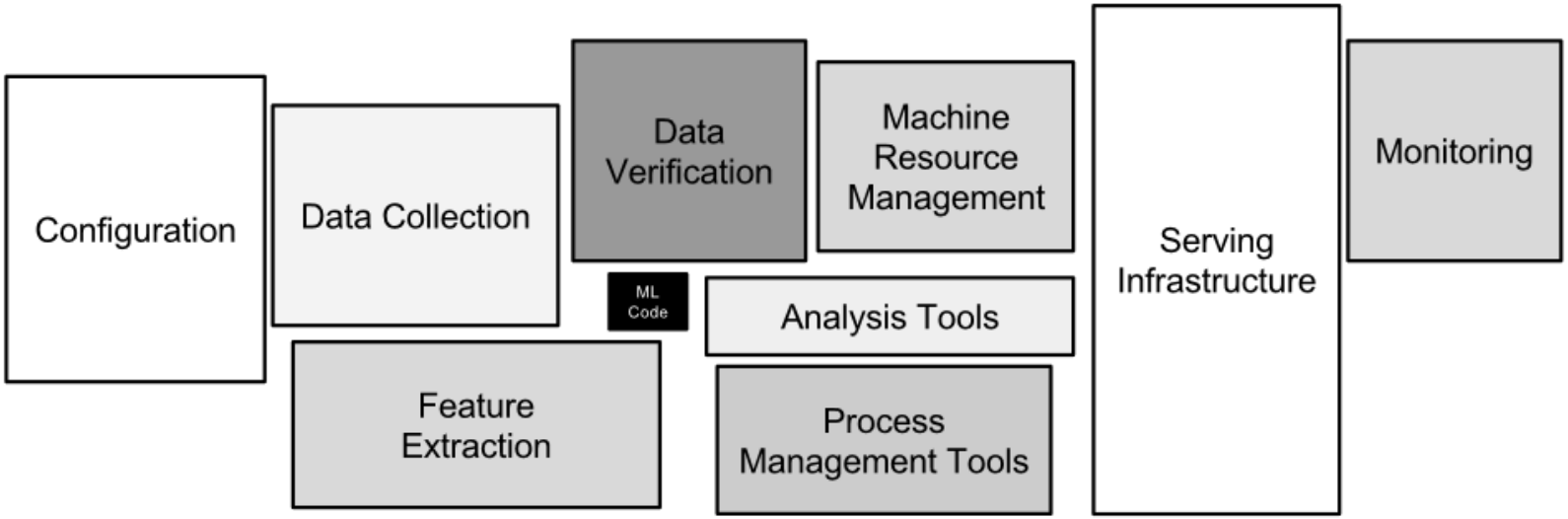


Why this course?

The gap between AI/ML research and production

An AI model is not a production system

- 90% of engineering time goes to data + infrastructure
- Model code is ≈**5%** of a production system



Sculley et al. “Hidden Technical Debt in Machine Learning Systems." NeurIPS 2015.

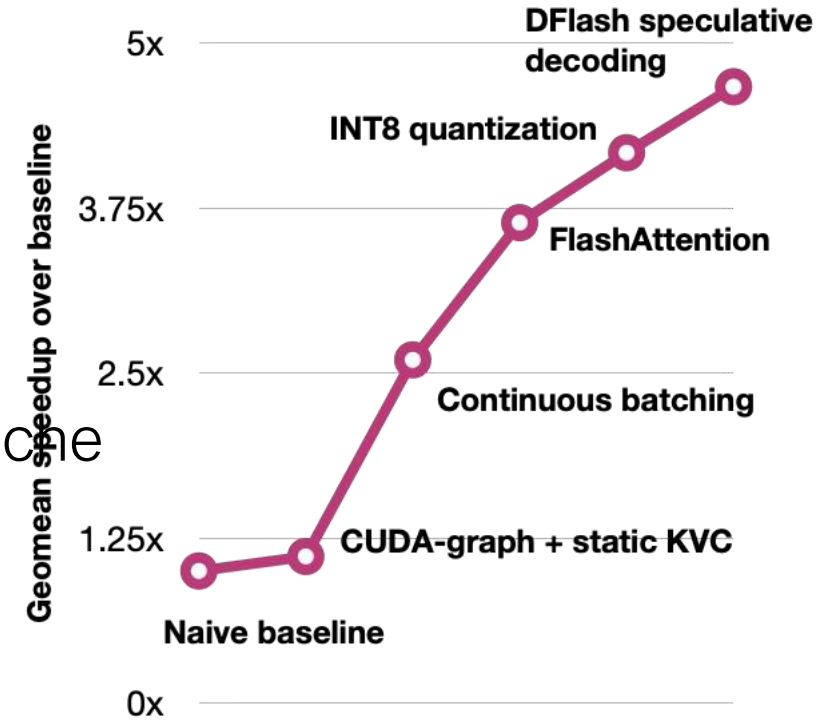
Why this course?

The gap between AI/ML research and production

Transformer: Algorithmic breakthrough

Autoregressive decoding:

1 token → 1 forward pass → reading model weights + KV cache



AI capabilities emerge from the **algorithms + systems** together.

Vaswami et al. “Attention is all you need.” 2017.

Why AI systems are different?

Software 2.0: Weights are the new source code

Software 1.0

- **Program:** explicit instructions
- **Bug:** trace to a line of code
- **Update:** edit the source file
- **Test:** deterministic paths

Software 2.0

- **Program:** learned weights
- **Bug:** trace to the training data
- **Update:** retrain on new data
- **Test:** monitor production distributions

Systems consequence: A Kimi K3-class model has ~2.8T parameters. The "program" is ~5.6 TB of storage. Debugging means auditing the data pipeline, not the Python code.

Karpathy. "Software 2.0." 2017. https://karpathy.medium.com/software-2-0-a64152b37c35

The bitter lesson: Scale beats expertise

Pattern across 70 years of AI:

- **1950s—70s:** Symbolic AI—logic rules, expert knowledge
- **1980s—90s:** Feature engineering—human-crafted representations
- **2000s—10s:** Statistical ML—learned features, manual tuning
- **2012—now:** Deep learning + massive compute **win**

The lesson: "General methods that leverage computation consistently outperform approaches that encode human expertise." – Sutton, 2019

Sutton. "The bitter lesson". 2019

The bitter lesson: Scale beats expertise

Pattern across 70 years of AI:

- **1950s—70s:** Symbolic AI—logic rules, expert knowledge
- **1980s—90s:** Feature engineering—human-crafted representations
- **2000s—10s:** Statistical ML—learned features, manual tuning
- **2012—now:** Deep learning + massive compute **win**

Systems implications: If scale determines AI progress, then the systems that move compute efficiently—data pipelines, training infra, hardware optimizations—are the limiting factor.

This course teaches the design and optimizations of AI systems

The lesson: "General methods that leverage computation consistently outperform approaches that encode human expertise." – Sutton, 2019

Sutton. "The bitter lesson". 2019

Course syllabus

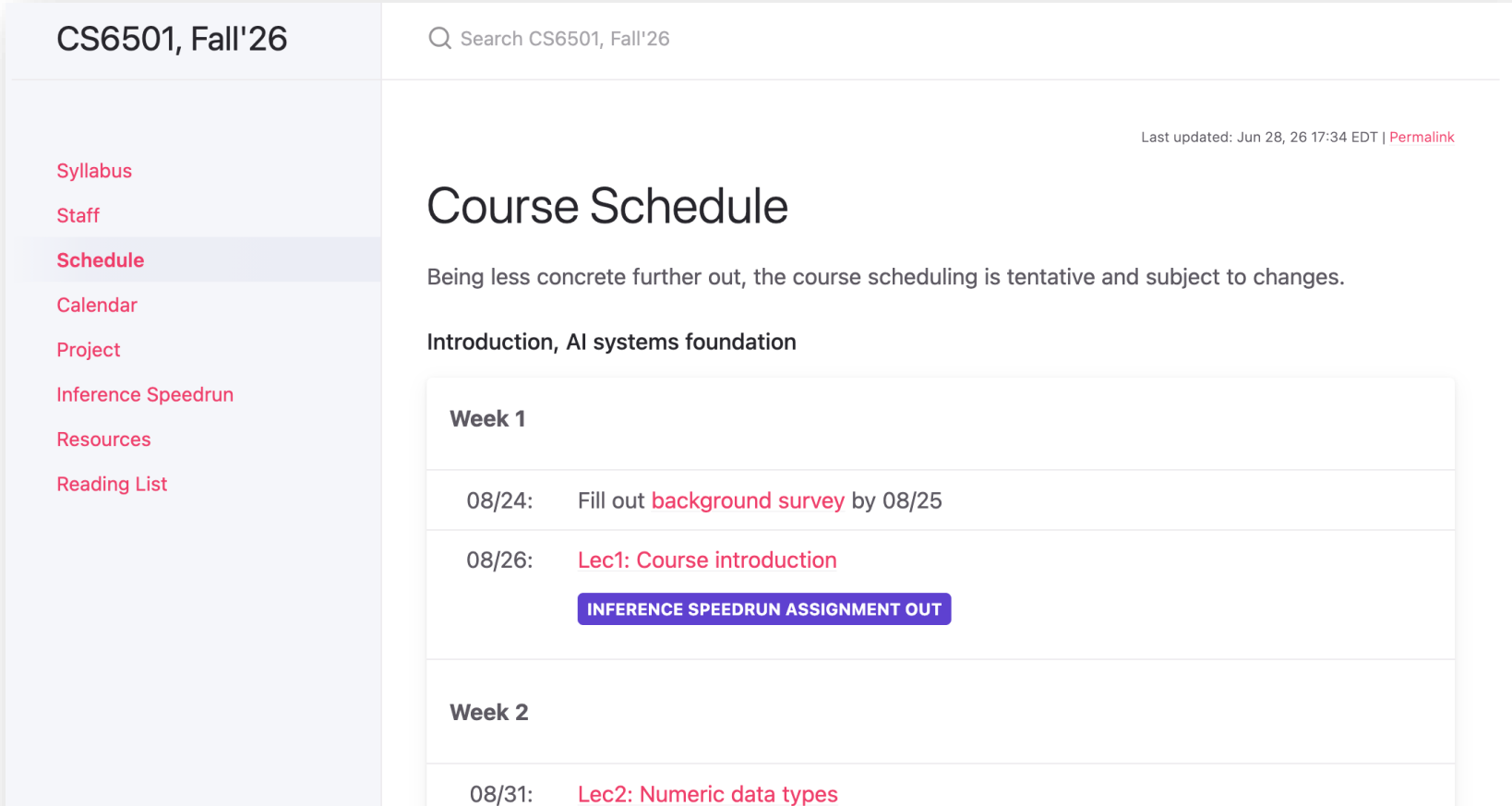
Big picture course goals

- Learn key concepts and techniques of modern AI (LLM) systems
- Learn how to read, present, and evaluate a systems research paper
- Gain hands-on experience building practical AI systems
  - You are encouraged to use your favorite AI coding agents/tools
- Learn how to approach, discuss, and communicate about technical subject matter

Schedule (tentative)

- Readings, project due dates, resources

  https://tddg.github.io/cs6501-ai-sys-fall26/



Schedule (tentative)

- Half of lectures led by me, another half led by one (or multiple) of you

- Week 1-3: AI systems foundation
- Week 4-5: AI infrastructure
- Week 6: Model training paradigms and perf engineering
- Week 7-8: LLM inference systems
- Week 9-14: Student paper presentations
- Week 15-16: Final project presentation

Readings

- Textbook (it's free!)
  - "**Machine Learning Systems (mlsysbook)**" by Vijay Janapa Reddi
- Papers/technical reports, occasionally blog articles
  - As most of recent topics are not directly covered by a text
- Slides/lecture notes

Paper reviews and presentations

- Two roles: Audience and Presenters

- **Audience:** Read at least one of two papers for that class (W9—14)
  - Ensure to understand key ideas, designs, findings
  - Fill the survey for that day (2 wild cards)
  - By semester end, you will have read 6+ quality AISys papers

- **Presenters:** Prepare slides and present the paper of your choice in class
  - With limited slots, some papers need >1 presenters
  - Presenter doesn't need to submit survey for that class
  - Send your slides to the instructor at least 2 days before the class

One semester-long assignment

- The only assignment in this course
- **LLM Inference Speedrun**

- AI coding tools are encouraged – and expected
  - But you must **understand what AI is doing, why it works, and why it fails**

- Milestones
  - **W4 (09/16):** Artifact and speedrun competition
  - **W5—W8:** Oral interview with the teaching staff
  - **W13:** Opinion+reflection report

Projects

- A term-long, research / development project
  - I'll supply ideas (and codebase for some)
  - You'll build the idea into an **MVP** (minimum viable product) with AI coding tools

- Week 4: Team signup (size up to two)
- Week 6: Proposal report due
- Week 10: Lightning checkpoint presentation
- Week 15-16: Project presentation + everything due

By the end of the semester...

Hopefully, you will have built some sophisticated, functional **MVP** that addresses an important real-world painpoint

Grading

- Assignment (20% total)
- Class participation (10%)
- Paper presentation (10%)
- Project (60%)

Resources and TODOs

- Sign-ups
  - Sign up for **Ed** (enrollment email sent)
  - Free cloud resources
    - Redeem $50 of **Google Cloud edu credit** – I'll send email
    - Microsoft Azure for students: $100 each with no credit card required: https://azure.microsoft.com/en-us/free/students

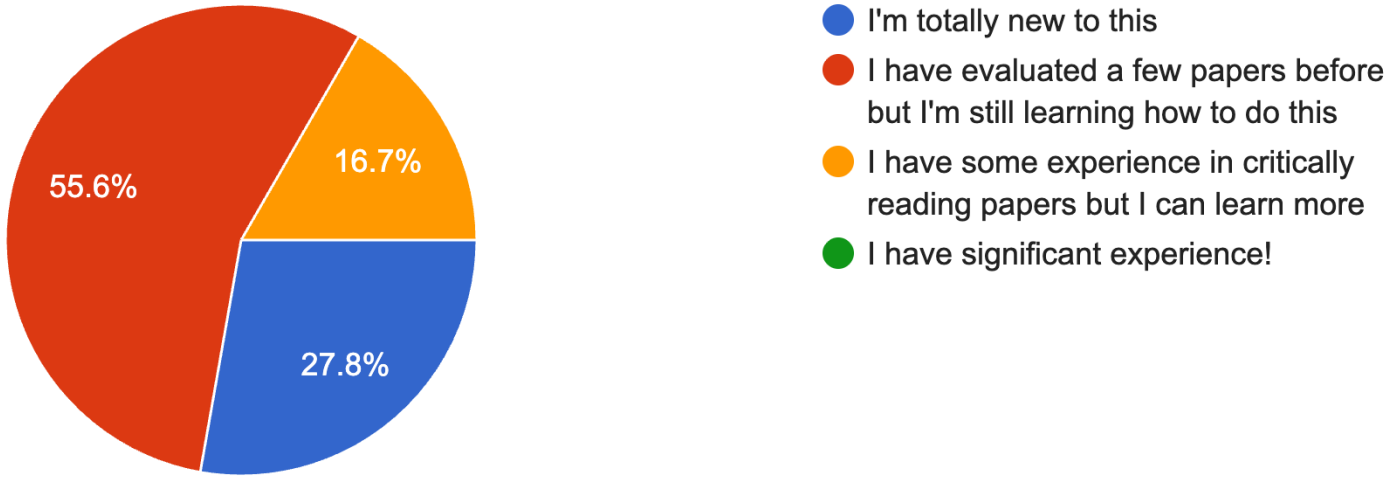
- Next week
  - Numeric data types and model compression

Background survey

Background survey

Q1: How much experience do you have in critically reading and evaluating systems research papers?

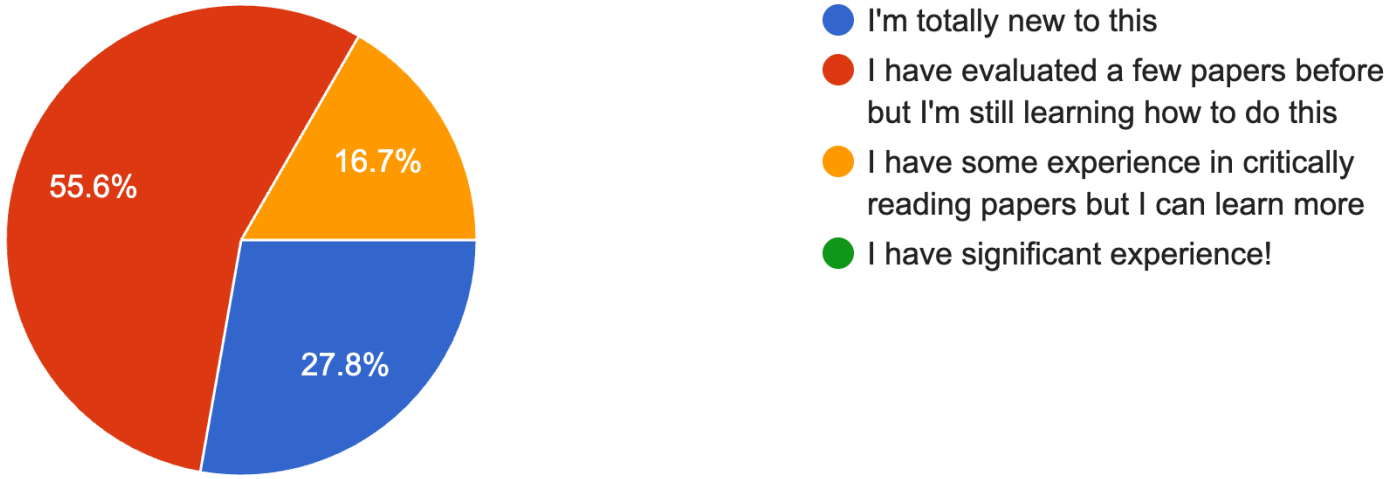
18 responses



Background survey

Q1: How much experience do you have in critically reading and evaluating systems research papers?

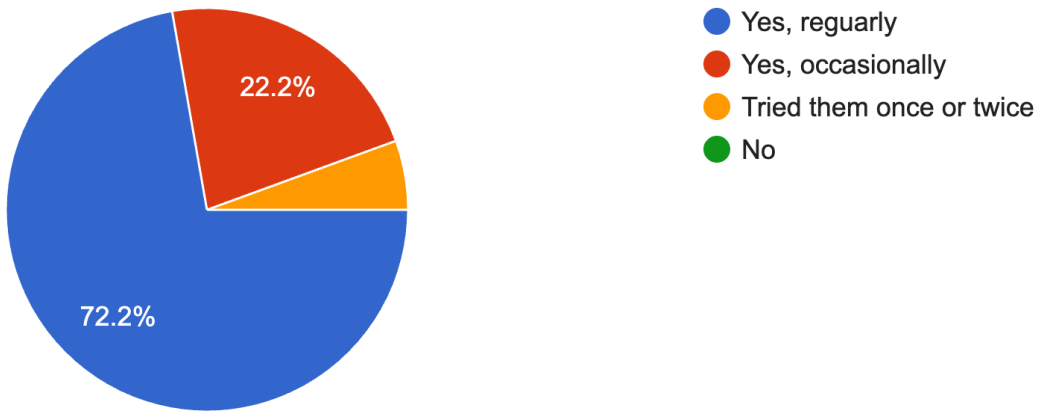
18 responses



- Best way to test if you understand a technique is by **teaching** it
- Best way to understand a technique is by **doing** it
- AI provides effective tools for smoothing the steep learning curve
- **Bonus credit: Reproduce any result or make a cool demo in your paper presentation and earn +0.5%**

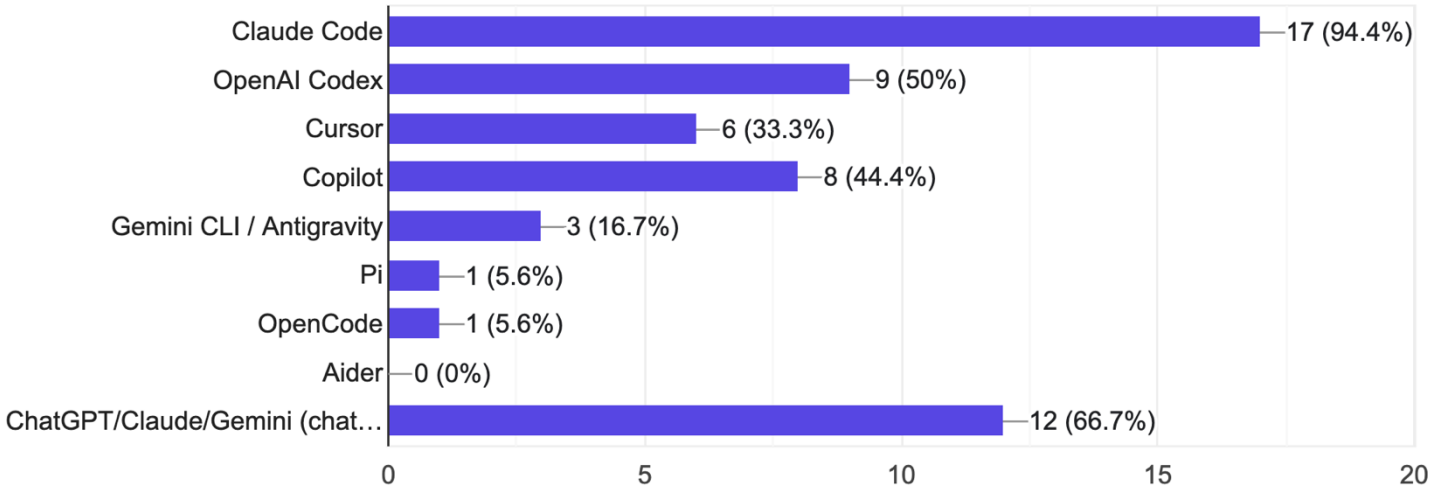
Q4: Have you ever used AI coding agents?

18 responses



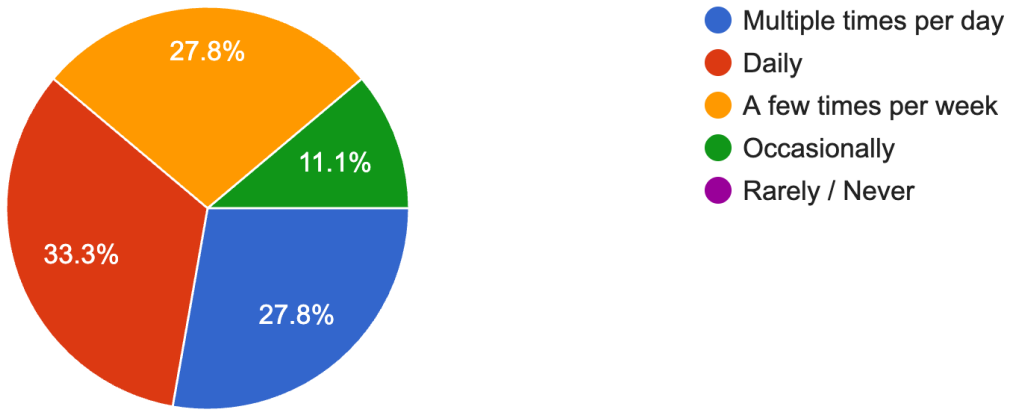
Q5: Which AI coding agents / tools have you used?

18 responses



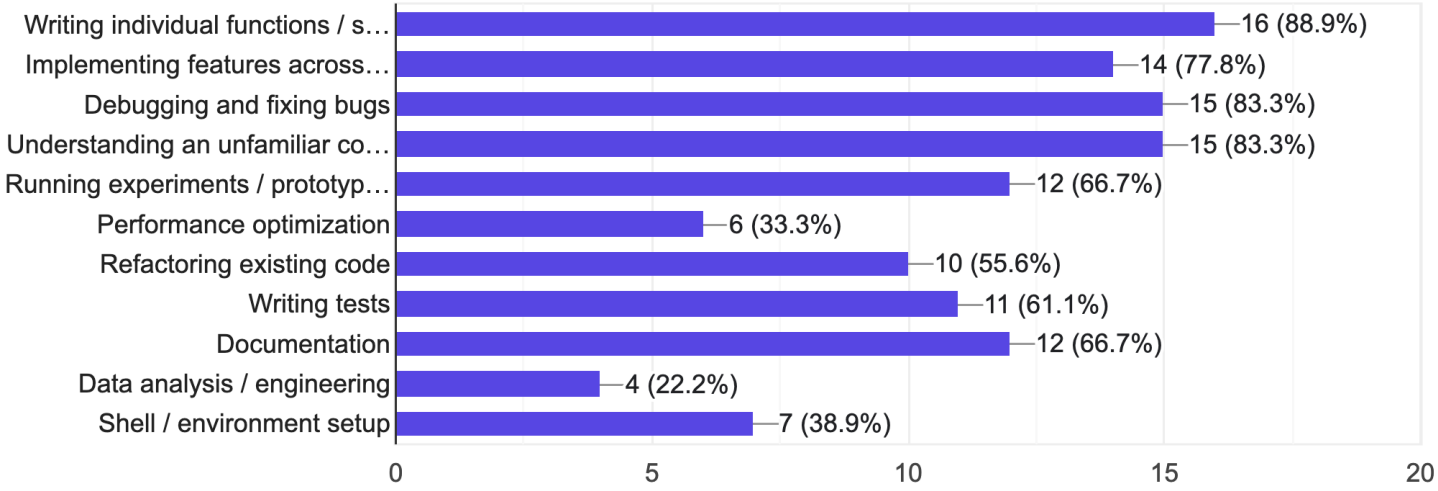
Q6: How often do you currently use AI coding agents?

18 responses



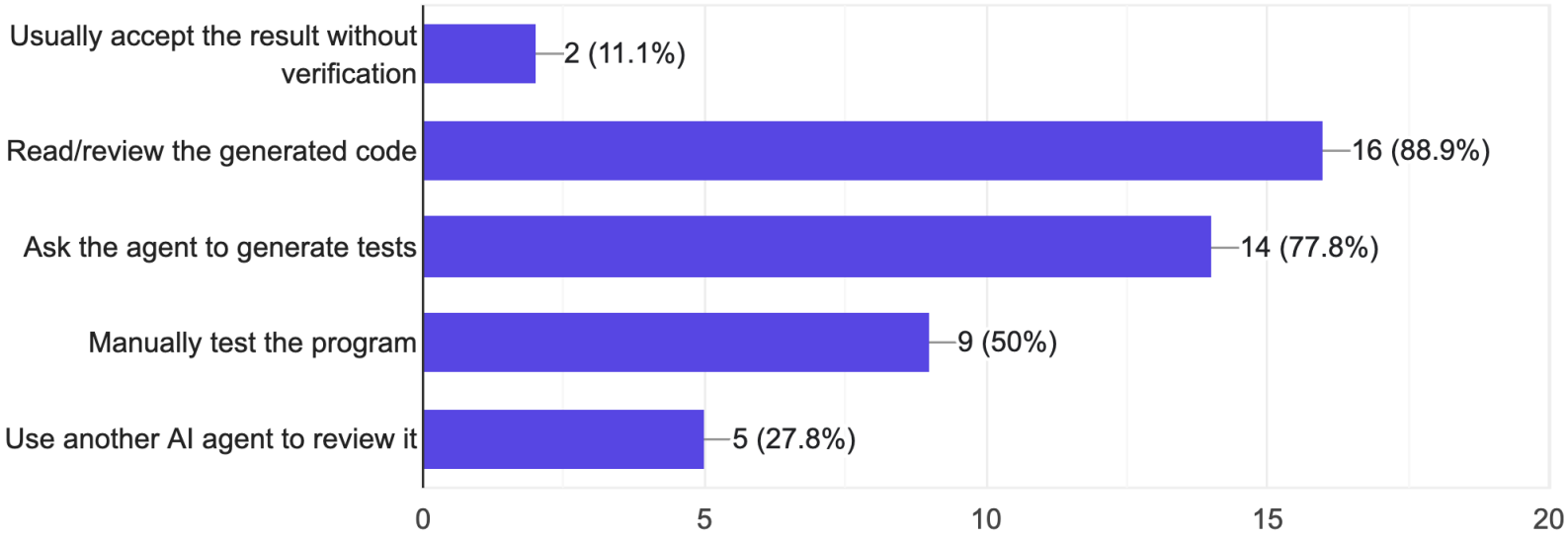
Q7: What do you use AI coding agents for?

18 responses



Q8: When an AI agent produces code, how do you usually verify it?

18 responses



Q9: What is the most complex task you have delegated to an AI coding agent?

- End-to-End Development and Architecture: Respondents have tasked agents with building full-stack web features, including front-end, API, and database integration, as well as designing programming languages with custom parsers and implementing complex fuzzing harnesses.
- Data Processing and Automation: Agents were utilized to perform end-to-end data pipelines, such as fetching, metadata tagging, chunking, and uploading documents to cloud storage, and extracting structured data from architectural floor plans for model evaluation.
- Testing and Quality Assurance: AI agents have been employed to manage the full experimental lifecycle—from writing and running tests to validating results—and conducting comprehensive audit reports to ensure feature parity when migrating legacy test suites.
- Technical Assistance and Debugging: Tasks included debugging unfamiliar concepts for website deployment, writing specific API endpoints, and assisting with the creation of research papers.

* Summarized by Gemini.

Time to introduce yourself