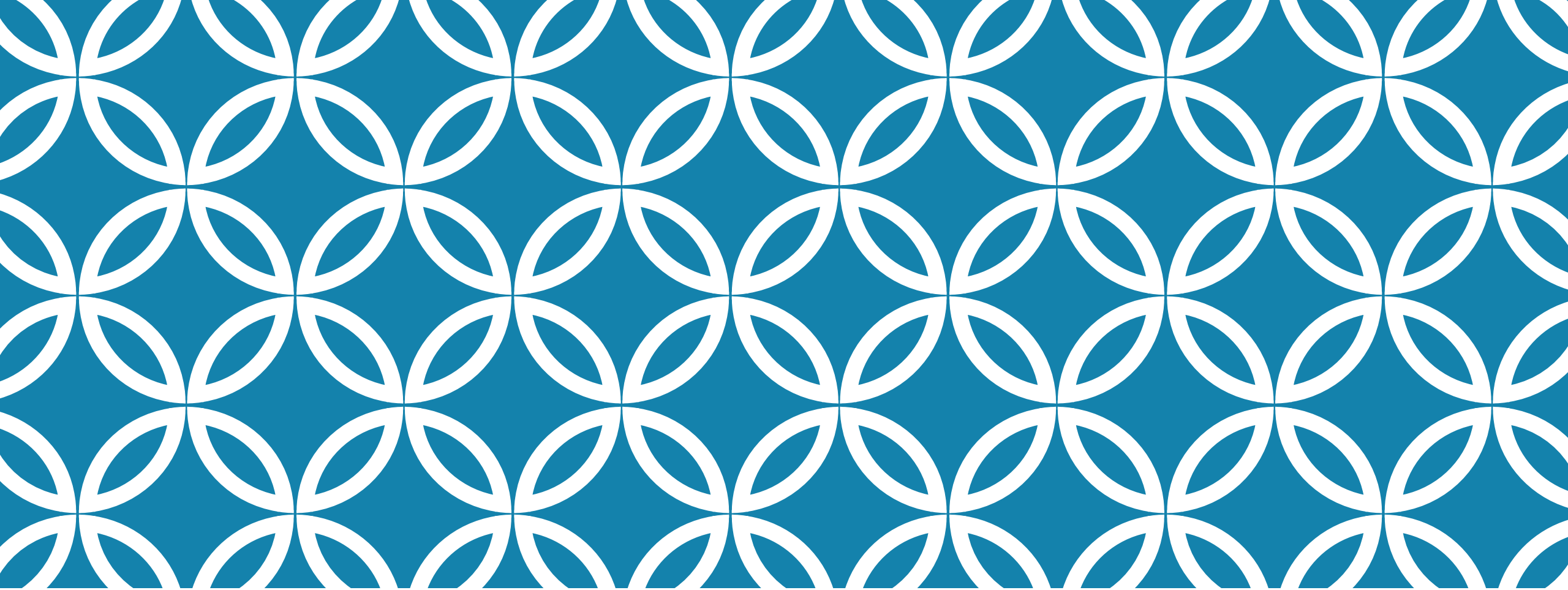


CS4414: SYSTEMS PROGRAMMING
CS5416: CLOUD COMPUTING AND ML HOSTING

Professor Ken Birman

Lecture 1

HOW CAN TWO COURSES SHARE MAIN LECTURES YET DIFFER?

Over the past years, CS4414 has enlarged to focus less on C++ programming (although we still teach that) and to include a glimpse of material from CS5412 (a discontinued course)

Meanwhile there is demand for CS5412, with its hands-on experience of distributed programming using the same tools employed on the cloud.

So we created this new fall course!

TWO COURSES IN ONE! FOR THIS TO WORK, RECITATIONS ARE REQUIRED, PART OF THE COURSE

CS4414 is for undergraduates, has a separate recitation section and its own grading formula. Satisfies the CS systems requirement. Homework is done individually.

CS5416 is for MEng students who have not previously taken CS4414. After the first programming projects, the remainder of the projects go their own way and involve learning to work in teams with cloud ML hosting tools.

TWO COURSES IN ONE! FOR THIS TO WORK, RECITATIONS ARE REQUIRED, PART OF THE COURSE

In all cases we require CS3410 (or an equivalent), because we assume experience programming with C, pointers and threads and with Linux memory virtualization models, covered in CS3410.

CS3410 cannot be taken concurrently, you must have completed it.

SUPPOSE THAT YOU ARE TRYING TO SPEED UP AN LLM TRAINING AND HOSTING SYSTEM

LLMs: Large language models. Gradually evolving into LRMs: Large reasoning models.

Many of them are multimodal (speech, vision, audio, …) and they may have lots of AI subsystems (a "mixture of experts" combined with vector databases and agentic subsystems).

Your boss is excited about SYSML: systems work aimed at efficient support for these massive jobs. She hired you to help on this!

HOW WOULD YOU GO ABOUT DOING THIS?

Probably wise to start by learning a little about LLMs and LRMs!

But the product code base will very likely be huge (maybe millions of lines of code), and some of the components could be proprietary (meaning: you can't even look at the code).

So… is it possible to speed up complex programs without knowing how they work?

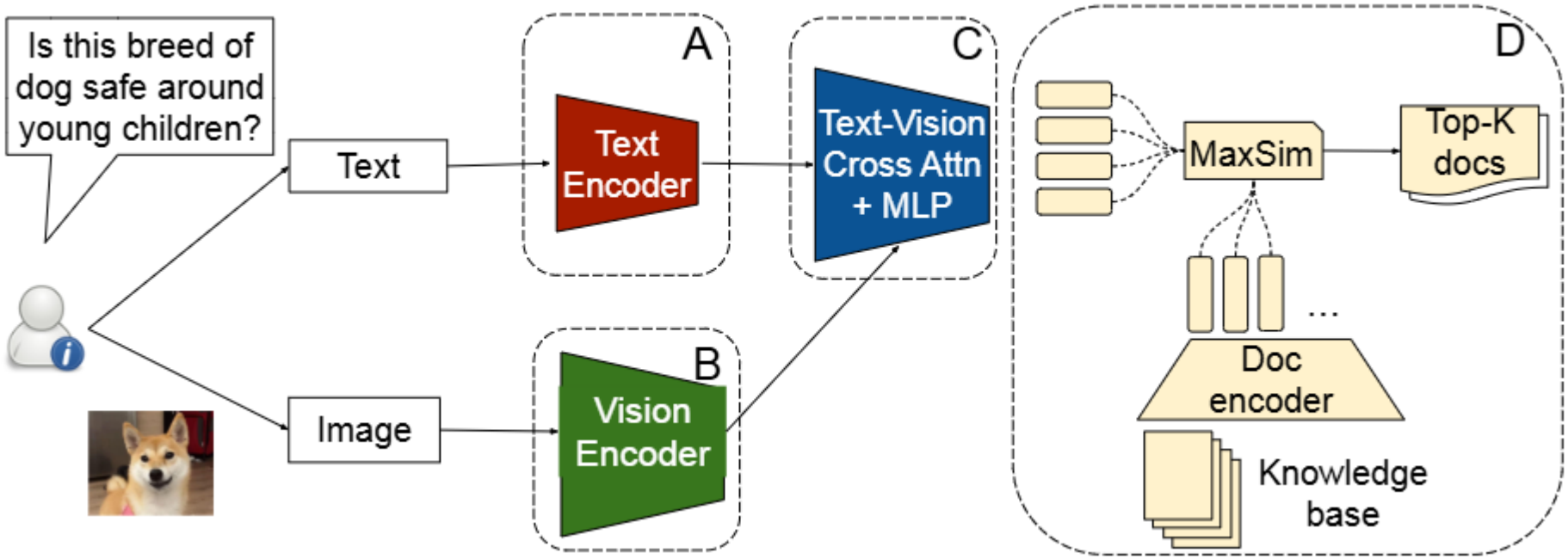
HOW DO LLMS AND LRMS WORK?

Prompts, images and other objects “flow in”.

They are converted to a universal representation: “embeddings” (vectors) in a high dimensional “information space”

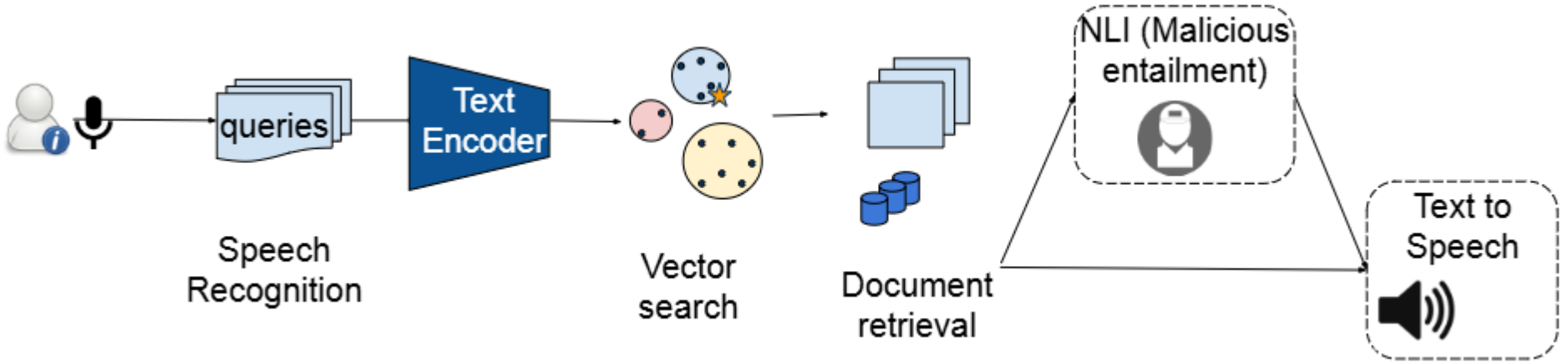
Related concepts form groups in this space… pattern match enables the LLM to map a query to the most-likely response.

A PIPELINE FOR ASKING QUESTIONS ABOUT IMAGES



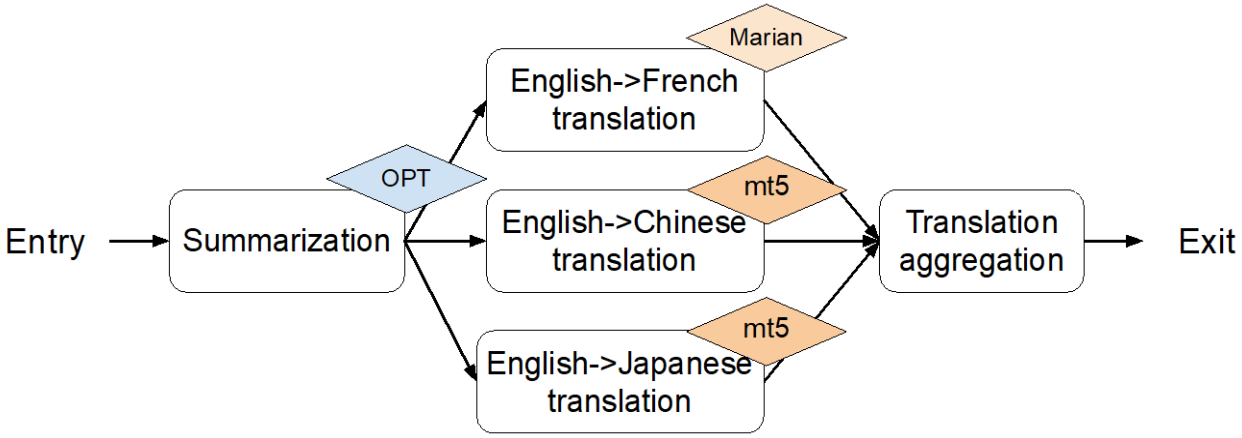
PreFMLR knowledge retrieval, finds documents relevant to a query.

USING DOCUMENTS ("KNOWLEDGE") TO RESPOND TO A QUERY

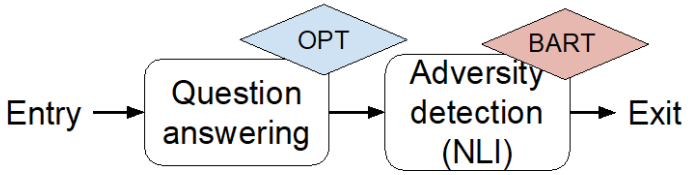


SpeechSense: audio Q/A for news feeds or document collections.

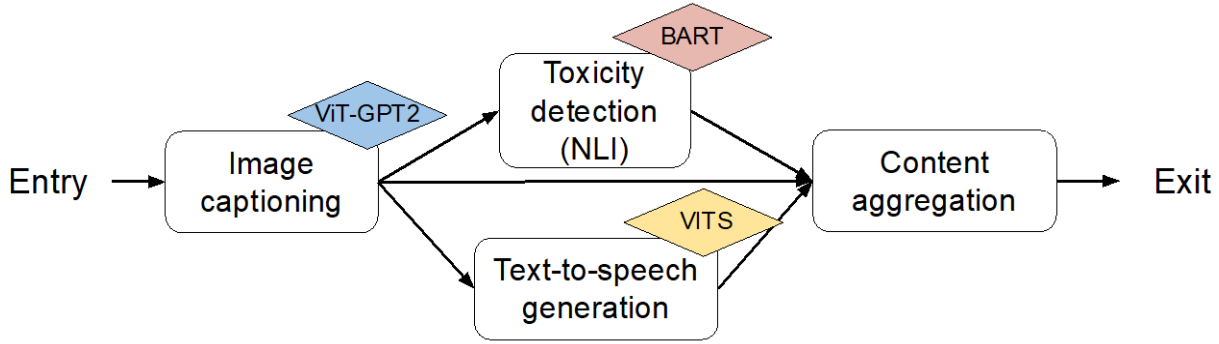
THESE THEN GET EMBEDDED INTO MORE COMPLEX APPLICATIONS, COPILOTS, ETC



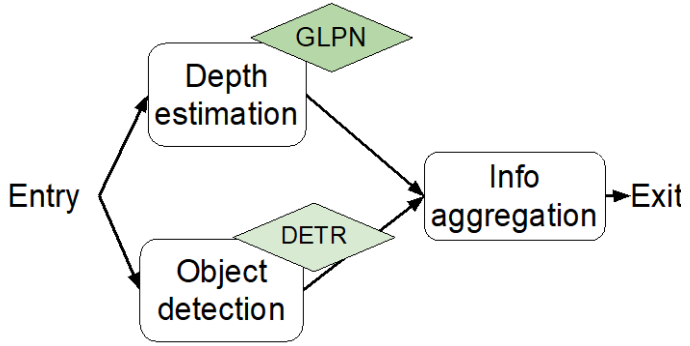
(a). Language translation pipeline



(c). Q&A dialogue pipeline



(b). Image reading pipeline



(d). 3D perception pipeline

SOME MIGHT NOT EVEN GENERATE WRITTEN OR SPOKEN RESPONSES

A copilot for code development generates coding suggestions

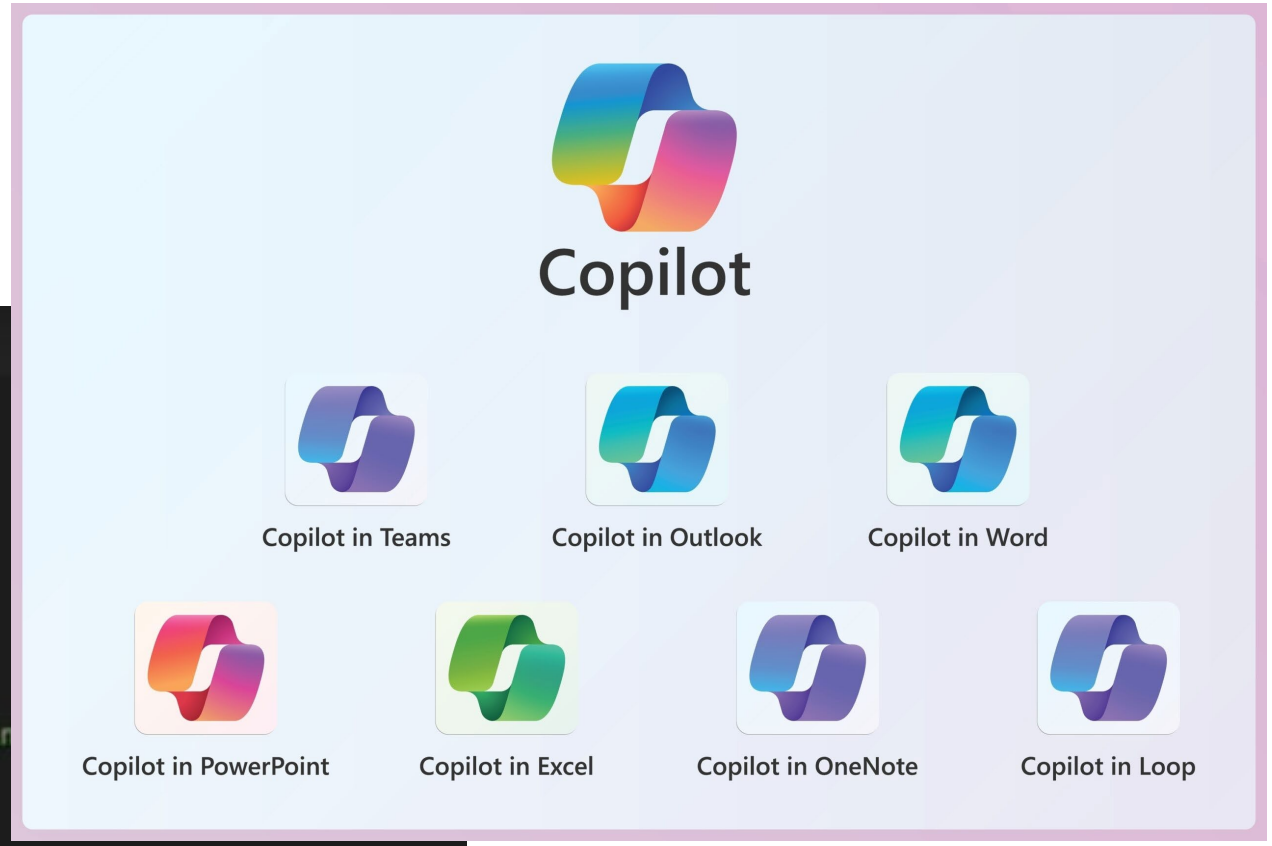
A physicians assistant might highlight an injury in a heads-up display so that a medical practitioner can “see” it more easily

An equipment servicing technician could be guided in her work using data integrated and interpreted by an AI

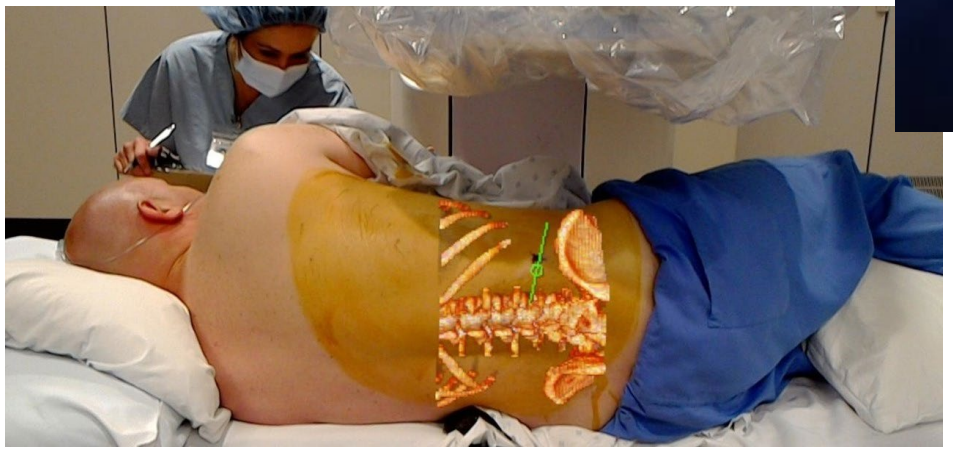
… COPILOTS

```
JS index.mjs 1 ●

JS index.mjs > ...
1   import {
2     Worker,
3     isMainThread,
4     parentPort,
5     workerData,
6   } from 'worker_threads';
7
8   // find all self-descriptive numbers in base 10 usi
9   if (isMainThread) {
10    const base = 10;
11    const max = base ** base;
12    const numThreads = 4;
13    const chunkSize = Math.floor(max / numThreads);
14    const threads = [];
15    for (let i = 0; i < numThreads; i++) {
16      const start = i * chunkSize;
17      const end = (i + 1) * chunkSize;
18      threads.push(new Worker(__filename, { workerData: { start, end } }));
19
20  // check if a number is self-descriptive in a given base
21  const isSelfDescriptiveNumber = (number, base) => {
22    const digits = number.toString(base).split("").map(Number);
```



… MEDICAL AR



… SERVICING EQUIPMENT

INSIDE THE BLACK BOXES

Typically we find deep neural networks pretrained on data, resulting in “inference” logic and a “model”

The model holds parameters representing neuron-to-neuron activation weights, and can be very large (hundreds of gigabytes or even terabytes).

Training is a slow, iterative process involving reinforcement learning: comparing model predictions to desired outputs, and adjusting if the model output is inaccurate.

THE CODE FOR ALL OF THIS REDUCES TO LINEAR ALGEBRA

Basically, matrix arithmetic (and dominated by matrix multiply)

But the matrices can be huge!

The space in which knowledge resides is high dimensional (it can be 1000 dimensions or more). Basically, we encode input, map it into this space, then remember it or do some form of ANN search

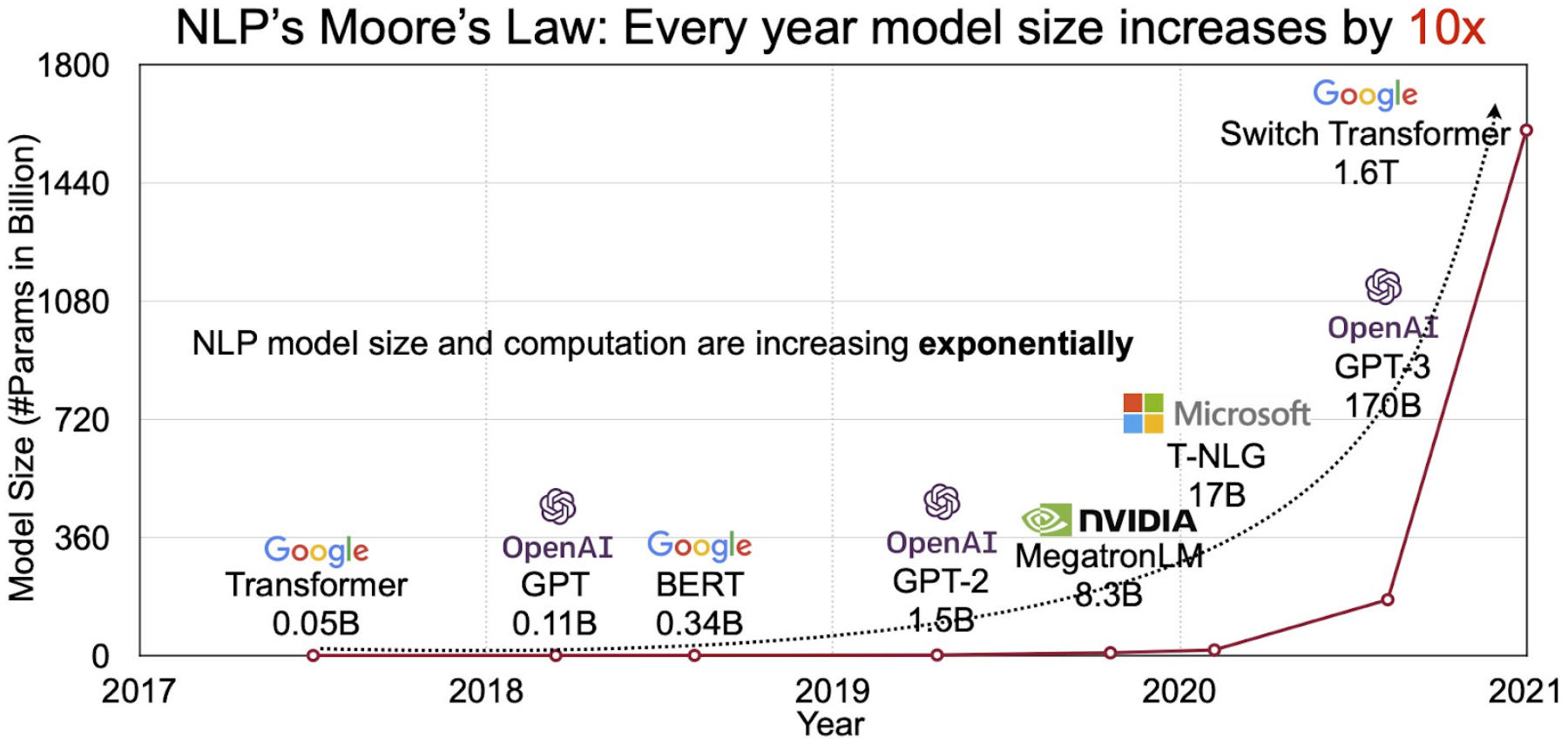
LLM/LRM TRAINING IS "AUTOREGRESSIVE"

They generate the response a token (or a few tokens) at a time.

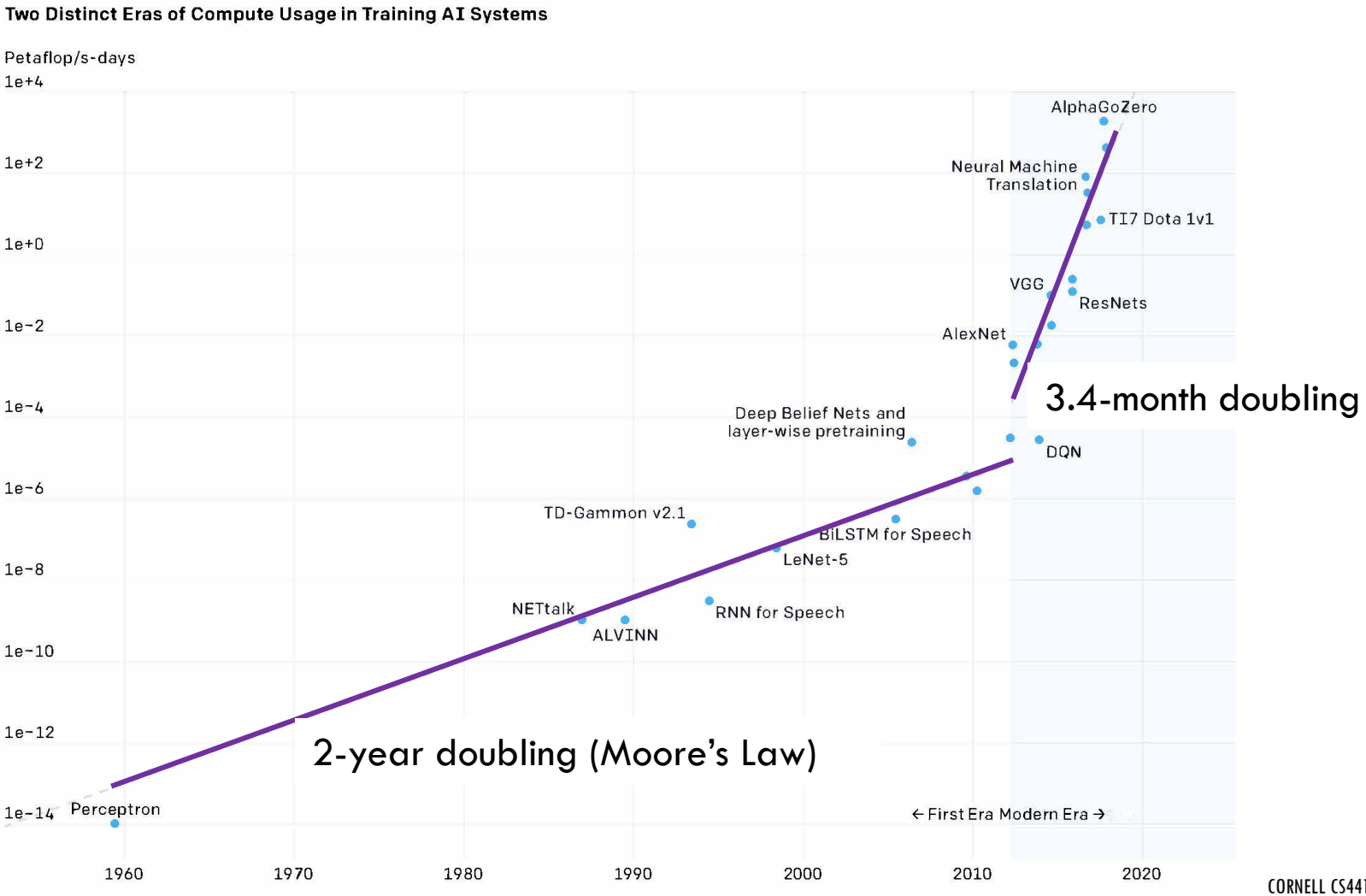
But then they assess the quality of the extension they are generating and will (1) explore multiple paths, (2) pick the best, and (3) roll back if somehow the generated text is of low quality

The output is used as a prior state when processing the next input.

MODELS ARE ALSO VERY LARGE, WHICH FORCES US TO USE MULTIPLE GPUS AND/OR MULTIPLE HOSTS



COMPUTE TIME TO TRAIN ML MODELS



YET (IRONICALLY), THE HARDWARE MIGHT NOT BE USED VERY EFFICIENTLY

Your boss has a good reason for concerns about performance.

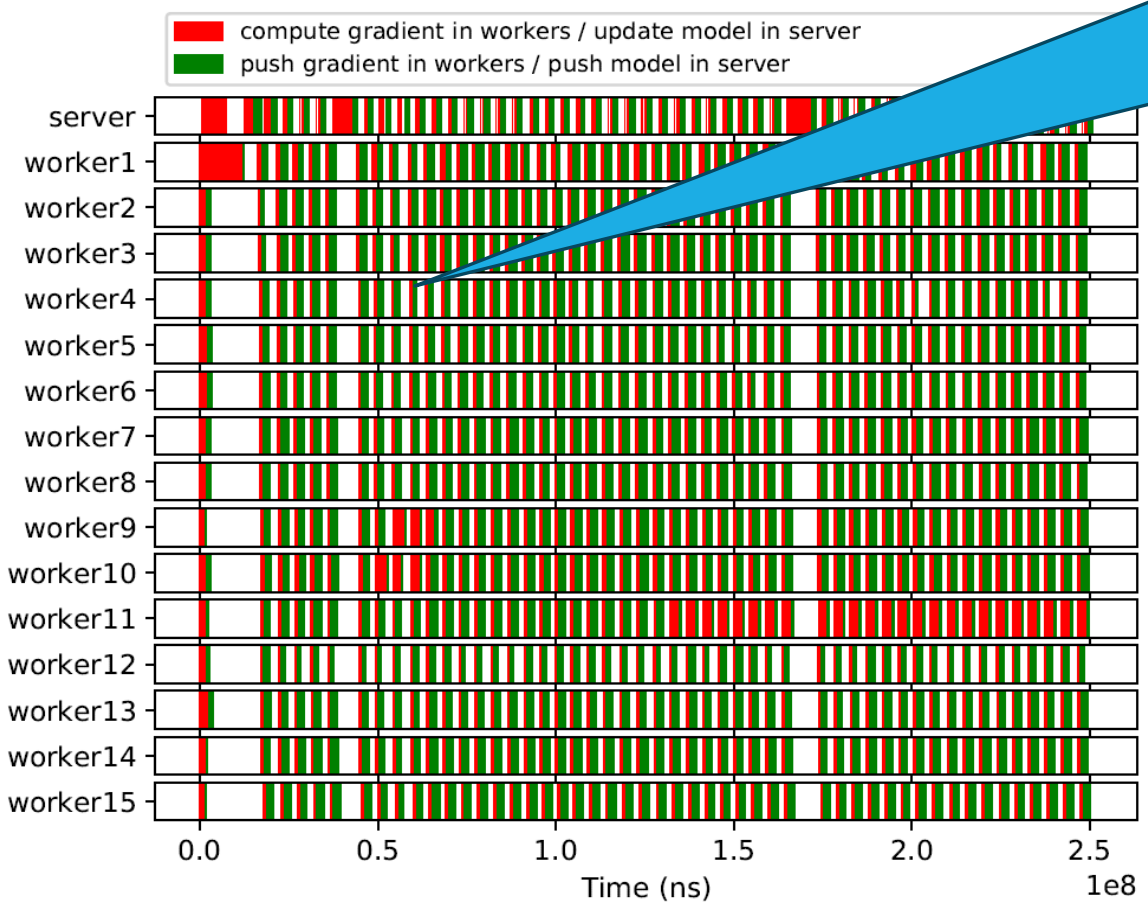
Training a specialized LLM or LRM can take thousands of hours of GPU compute time (literally, days). Very expensive!

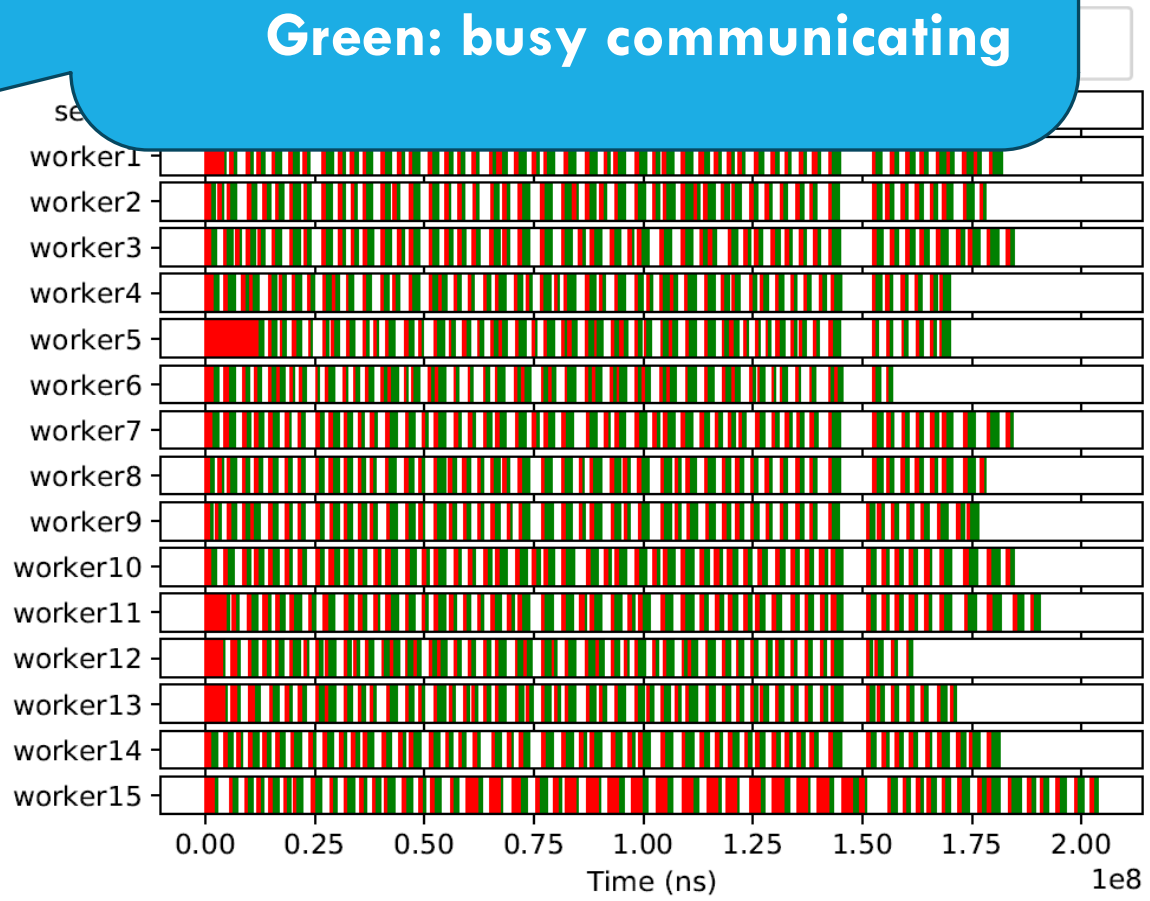
Yet because these training systems and inference tools are being built in such haste and evolving so fast, they are poorly optimized

SGD SCENARIO WITH 16
GRAPHS BY KWANGHYU

Key:

Red is compute
White is idle
Green: busy communicating





… KWANG'S SGD EXAMPLE WAS A HOME-MADE SCENARIO. PRODUCTS ARE BETTER… RIGHT?

The graph Kwang made was for a home-brew SGD training system that wasn't terribly efficient

But on the other hand, his visualization raises a question: are people routinely tuning performance of production solutions?

And the answer seems to be: Not systematically.

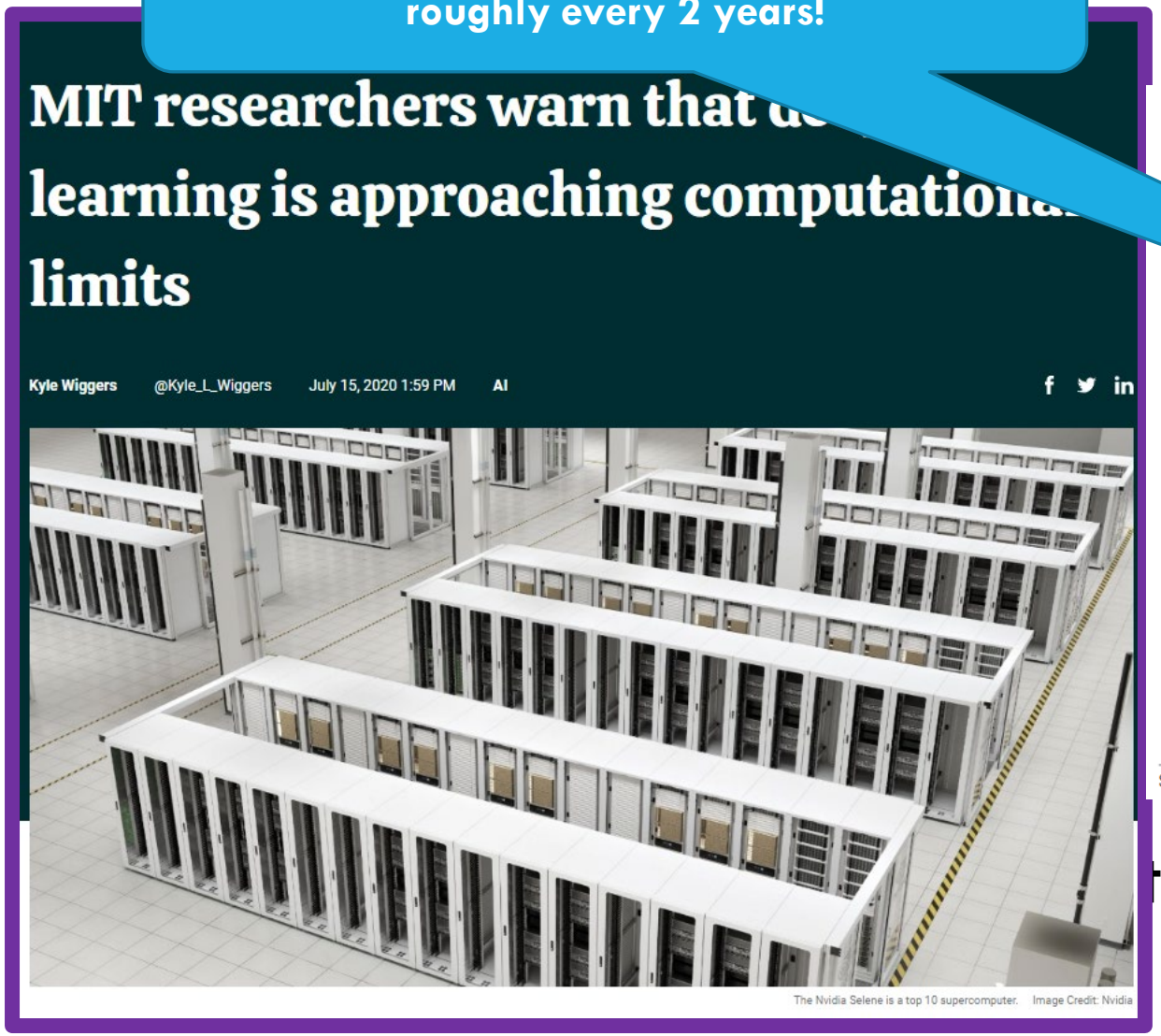
THIS IS CAUSING A GLOBAL ENERGY CRISIS!

In the US electric power is becoming more expensive in part because households are now competing with AI (and yes, crypto mining) data centers for a limited resource

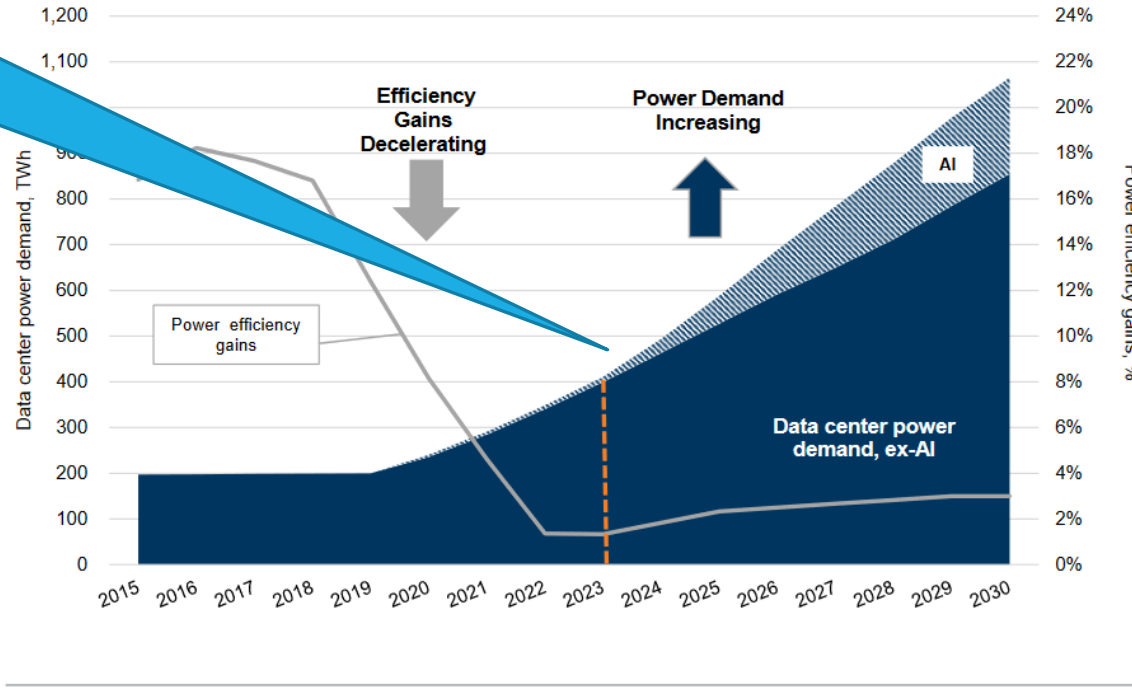
In some regions power is still produced mostly from coal: a very polluting fuel. In others hydroelectric dominates, but new dams are ecologically impactful and not always in good ways

Another option is to build more nuclear power plants, or perhaps to bet on a pivot to fusion… but fusion isn't yet available

Roughly 1% of global electric use, doubling roughly every 2 years!



https://venturebeat.com July 15, 2020



Source: Masanet et al. (2020), Cisco, IEA, Goldman Sachs Global Investment Research

on whether your code can
t consumes!

https://energyinnovation.org/2020/03/20/how-much-energy-do-data-centers-really-use/

SMALLER GENUINELY CAN BE BETTER

Many AI computational tasks scale “superlinearly”, meaning double the size of the inputs costs more than double the compute

So it makes sense to try and break these massive LLM and LRM models into smaller ones, using various forms of supervised learning (a big model trains a smaller model)

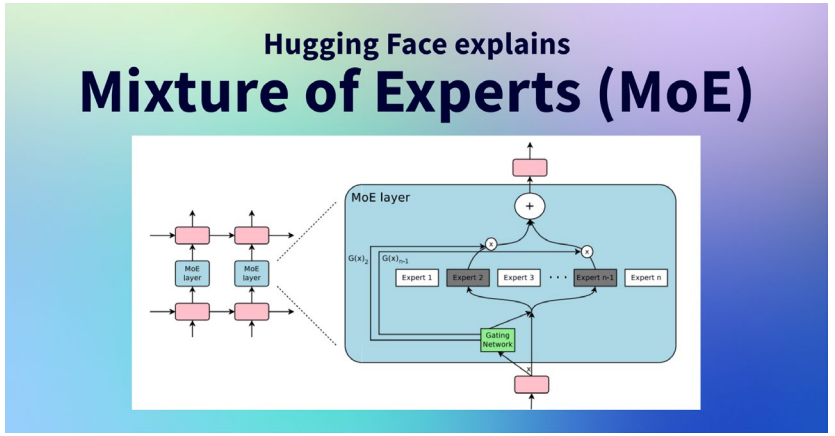
MOVING TOWARDS MULTIPLE PROGRAMS

There is nothing about the logic flow that forces these systems to be distributed. It happens because of modularity and big data objects.

As recently as a few years ago, LLMs and LRMs mostly ran as monolithic programs, but researchers found that different scenarios really require different expertise.

For example, dialog about Taylor Swift is very different from dialog about math problems, or about asthma. "Specialist" LLMs do better.

MIXTURE OF EXPERT LLMS



Around 2021 leading companies began to train multiple models to cooperate, each with a different expertise (different training data).

Then they could classify each query based on which experts would be most relevant, and to what degree (what weight each deserves). They treated response generation as a kind of expert, too.

A weighted mixture of experts enabled rapid progress and the individual models could be smaller. But trillion-parameter models have emerged in the area of LRMs.

IS A MIXTURE OF EXPERTS ALWAYS A DISTRIBUTED SYSTEM?

Not necessarily! It could still be implemented as a single program that just has multiple subsystems running (in parallel).

… but there are many mixture of expert systems that do run as distributed programs, for lots of reasons.

ADDITIONALLY, LLMS AND LRMS TALK TO OTHER CLOUD HOSTED SERVICES

Many (in fact, most) problems require fetching documents from repositories that seem closely matched to the task:

- Big collections of files – perhaps *very* big
- Databases
- A specialized kind of database called a "vector" database. These are used for approximate match: *find me documents similar to this one, or that have content related to this query.*

WHY DOES THIS MAKE THEM DISTRIBUTED?

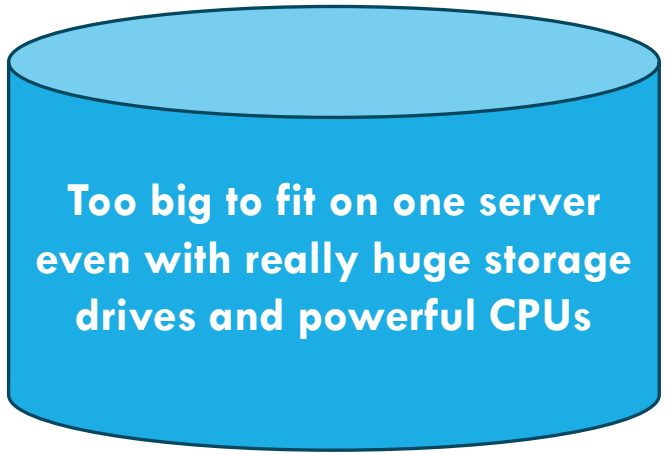
Each of these big roles tends to be handled by a specialized and highly optimized big computer subsystem, like a remote file system on a cloud (think of google docs or dropbox).

Databases and vector databases are very specialized and complex, huge code bases. And very valuable property to the companies that create them.

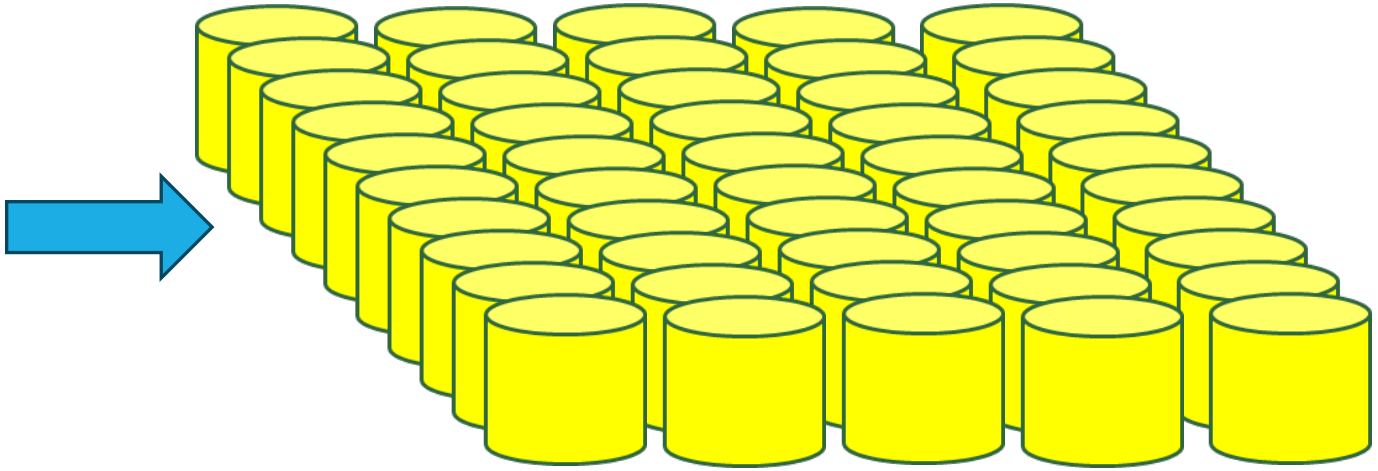
And many compute tasks in these LLMs and LRMs need parallelism

FOR EXAMPLE, REALLY BIG DATA IS SHARDED

Shard: refers to a "slice" or "portion" of a bigger data set



Old approach: Buy a big database server



Modern approach: Buy lots of medium sized database servers

MORE REASONS…

Images or audio require a different approach to initial data analysis than text. The code will tend to be created by different teams… and will run as separate programs

Sometimes our goal is to generate text, but other uses involve prediction of what the user is doing, knowledge retrieval, classification of a request among a set of categories, generating an image or video…

… BUT THIS IS NOT ALWAYS THE CASE

For example, many people who use PyTorch really like big public repositories of code and data, like from Hugging Face

For them, AI coding tends to look just like high level function calls and composition. Call A with some input, pass the output to B…

A and B might be small… might be huge. The AI developer is a busy person and probably won't care (unless forced to!)

SO… AI IS…

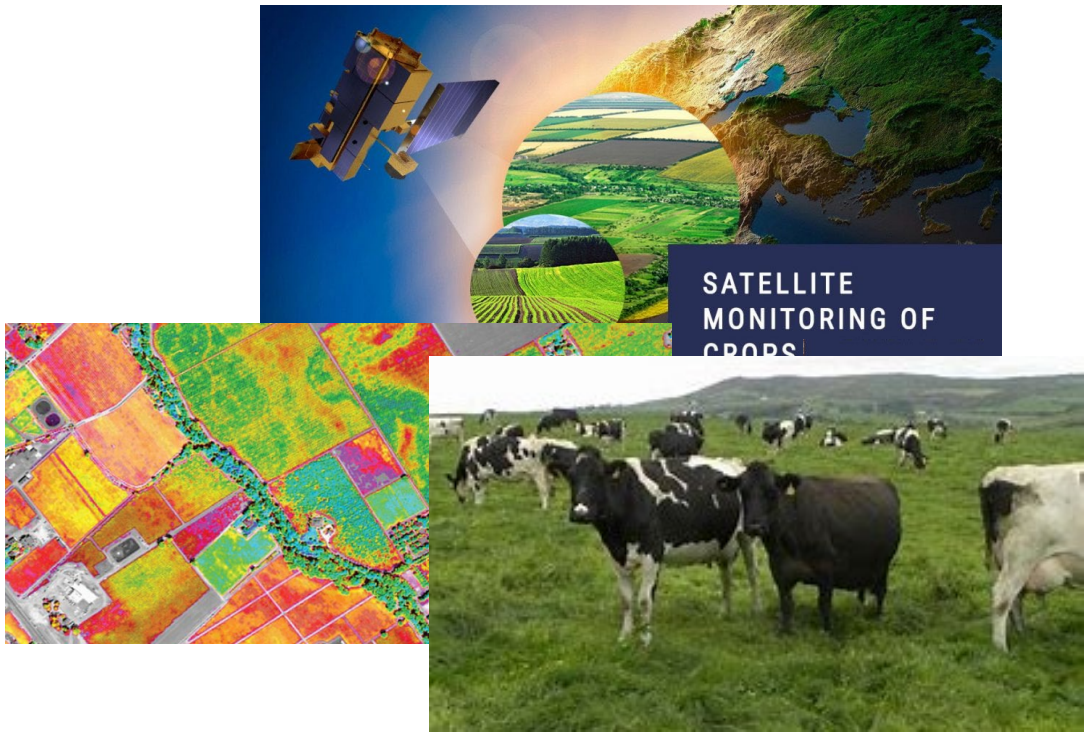
A mix of huge single programs, distributed programs, remote services accessed on the network.

Lots of things use GPU accelerators (or other options like LPUs)

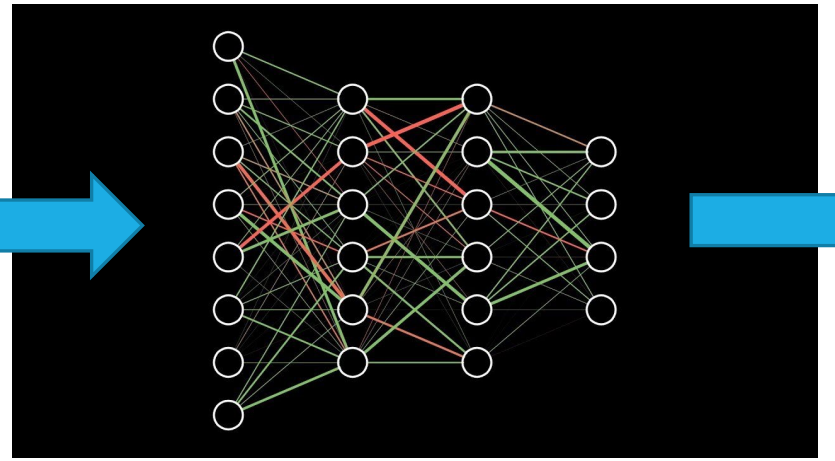
A huge, complicated, demanding mix. Often “nobody” has the role of understanding and tuning performance.

ML IS BEING EMBEDDED INTO A SURPRISING RANGE OF SETTINGS… LIKE SMART FARMING

Data sources



Cooperative ML
Distributed AI



Smart
Queries

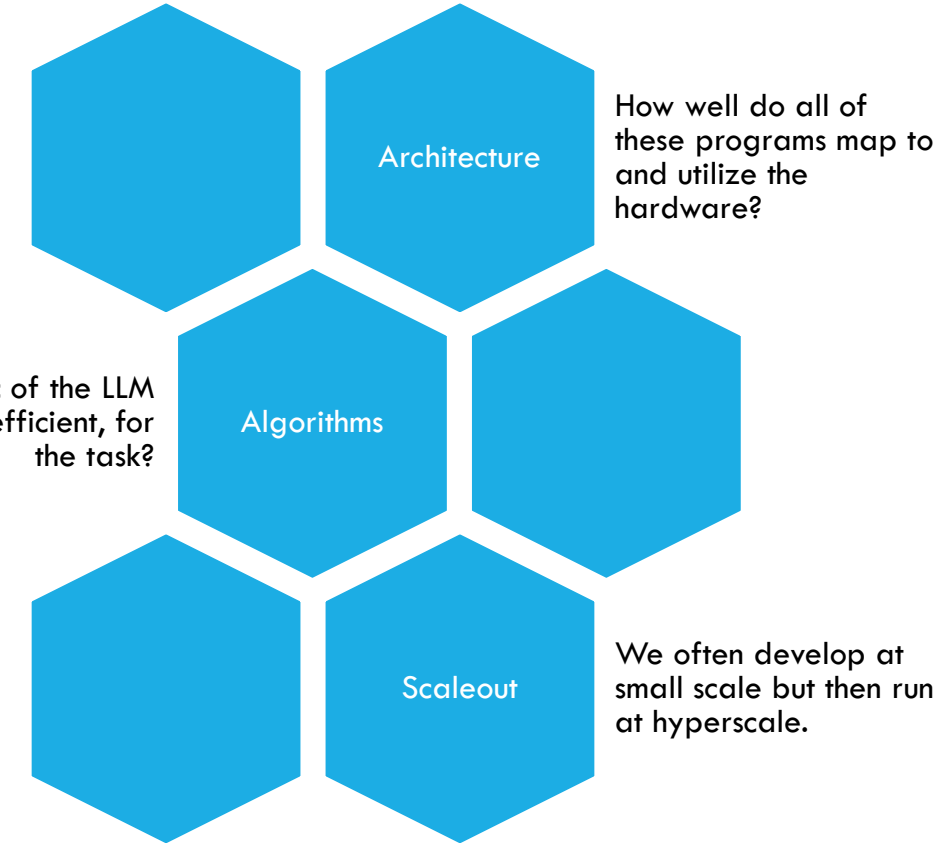
How much should I budget for raw milk purchases in March for my cheese factory?"

… THERE CAN BE MANY AI COMPONENTS IN THESE PIPELINES

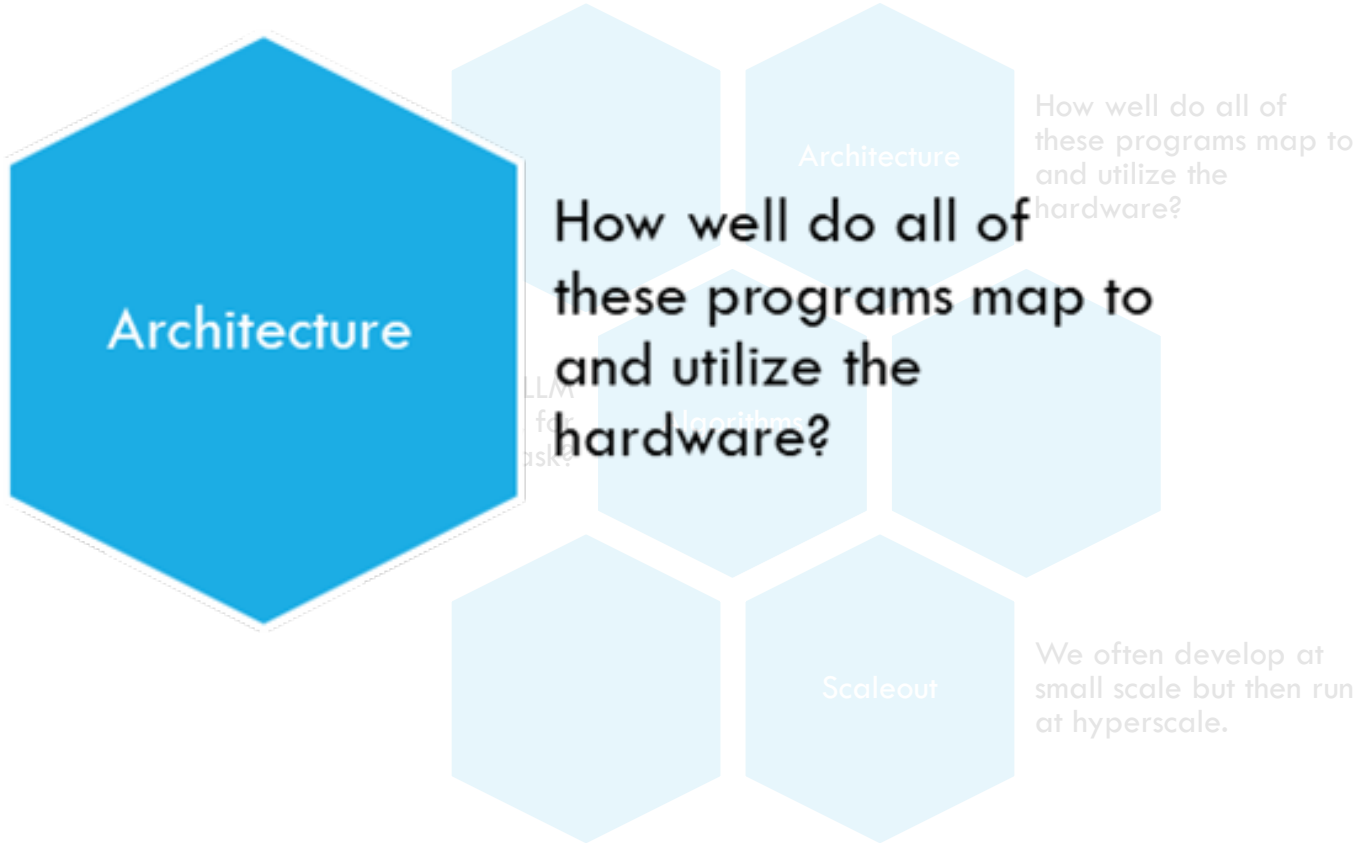
For example, updates and queries for one approach employ:

- Document chunking, using a transformer
- Document and query embedding, again via a transformer
- Formation of point-clouds representing clusters
- Approximate nearest neighbor search
- Enlarged prompt formation
- Actual generative response (and sanity/toxicity check)

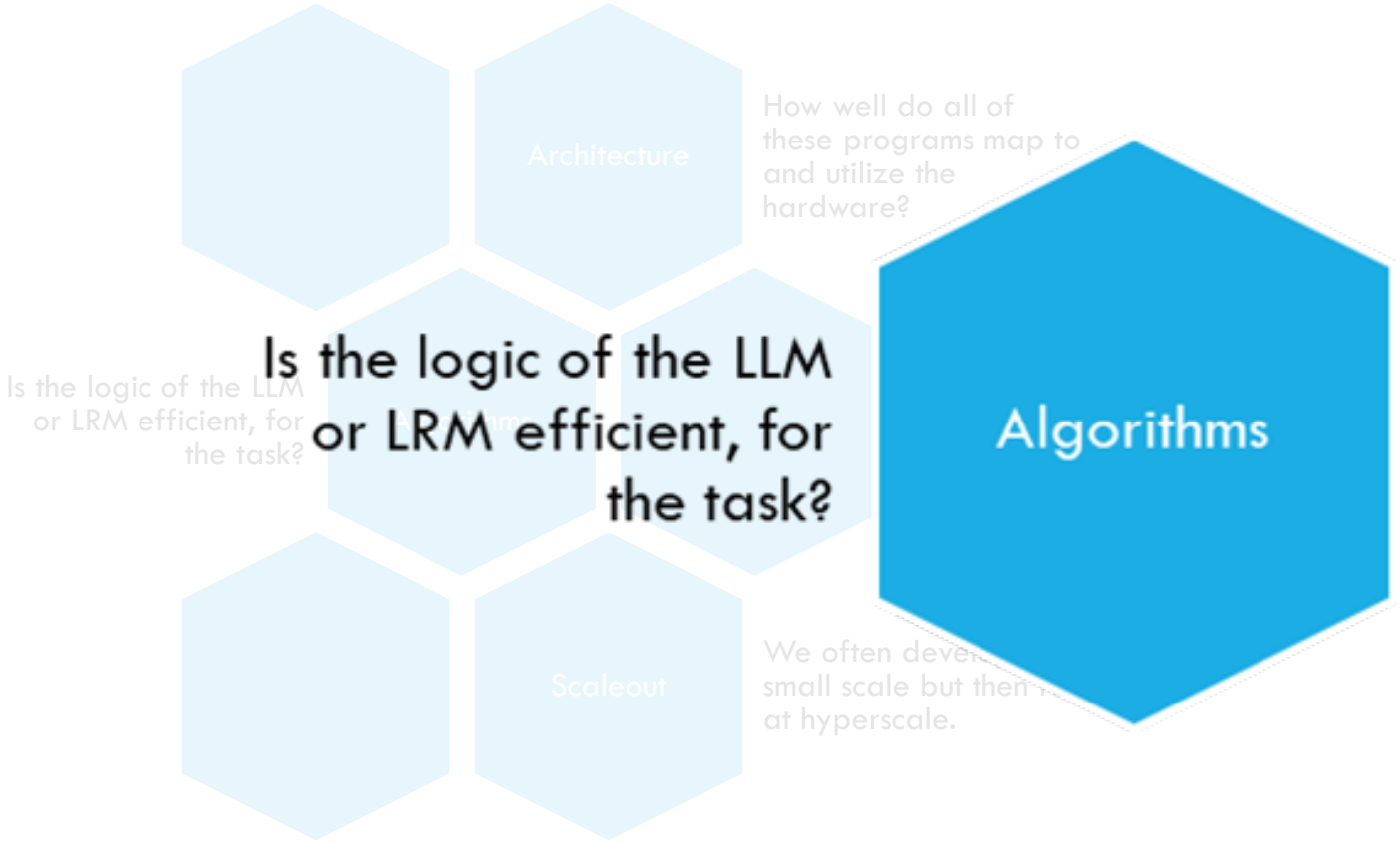
WHEN WE THINK ABOUT PERFORMANCE, THERE ARE MANY DIMENSIONS TO CONSIDER!



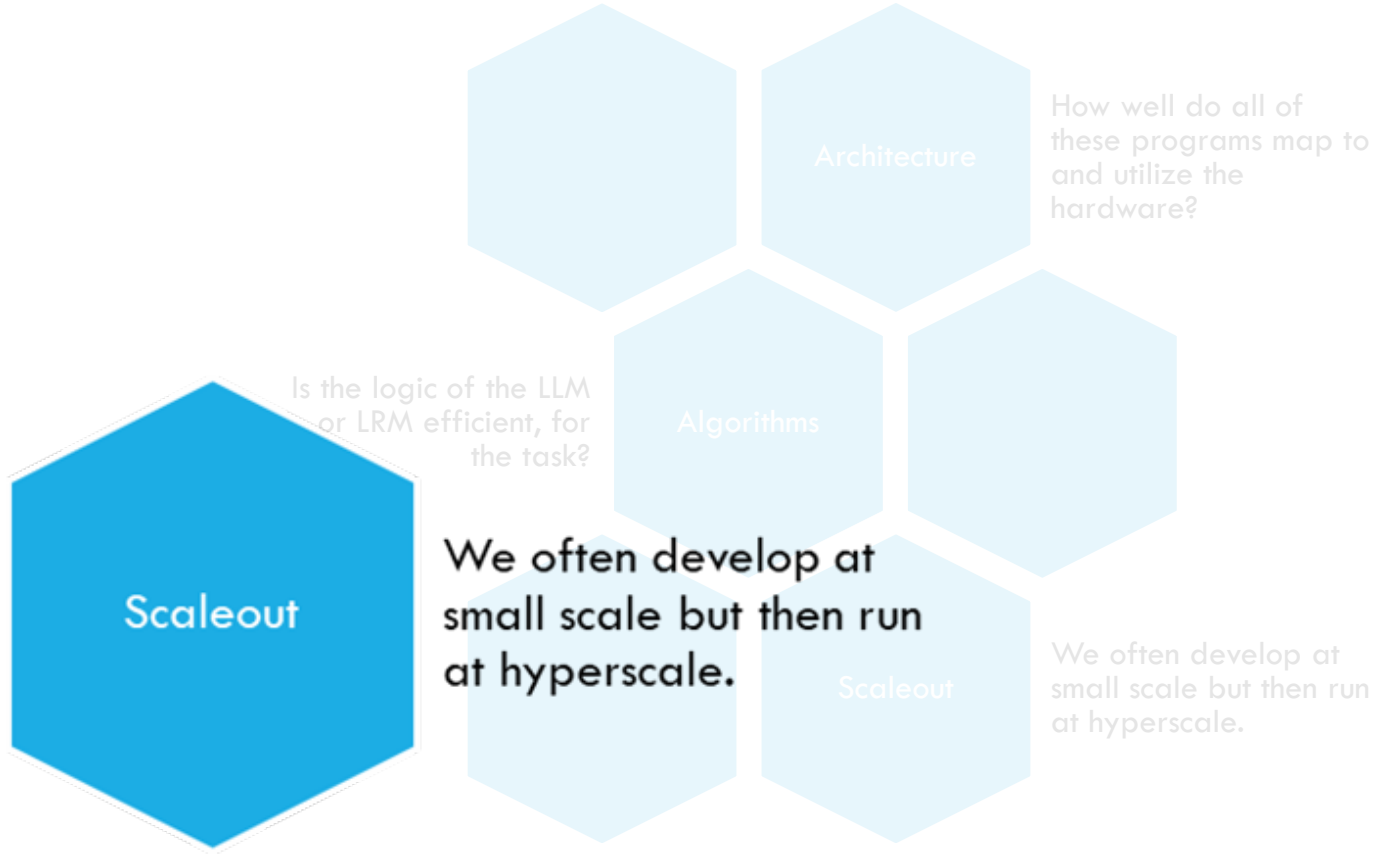
WHEN WE THINK ABOUT PERFORMANCE, THERE ARE MANY DIMENSIONS TO CONSIDER!



WHEN WE THINK ABOUT PERFORMANCE, THERE ARE MANY DIMENSIONS TO CONSIDER!



WHEN WE THINK ABOUT PERFORMANCE, THERE ARE MANY DIMENSIONS TO CONSIDER!



EXAMPLES

Suppose that our system needs to fetch lots of big objects over the network. It would pause frequently. *Can we do better?*

Or what if the AI is designed to distribute work over multiple workers (perhaps, each with its own GPU). Are they all working concurrently or is there a lot of pausing to wait? *Can we maximize parallelism? Is the hardware maximally busy?*

SOME OF THESE ARE ALGORITHMIC ISSUES

In CS you've learned about the power of better algorithms

But once a system is coded, issues like asymptotic scalability shift to questions of practical efficiency

Remember Kwang's experiment? What if these massive, costly AIs are full of those kinds of inefficiencies?

PAUSE HERE FOR CLASS DISCUSSION

I want to hear some of YOUR ideas for

- Situations in which complicated systems might be inefficient,
- Verifying our guesses (which may sometimes be wrong),
- Improving performance without rewrite the code.

A LOT OF IDEAS!

In our course, we need to start by learning more about what makes programs (and distributed systems) perform really well.

- First, at small scale: one thread on one CPU
- Then inside a typical NUMA compute server (multithread parallelism)
- And then distributed over a set of servers and GPUS
- And finally, interacting with the O/S and various services

AT FIRST, WE'LL FOCUS ON PROGRAMS YOU MIGHT BUILD FROM SCRATCH

Understanding performance is a very hands-on talent!

It requires coding with the architecture of the computer in mind, being "compiler-friendly" to get the maximum from how it maps code to instructions, and then repeating this for networked code.

So our journey starts with "one program" on "one NUMA server"

… BUT THEN WE'LL EMBRACE SCALE

Your boss at that future job is using code and systems from many open source projects and combining that with licensed products

You won't be in a position to rebuild those solutions! You*might* be able to modify the platforms they run on.

So at large scale, with complex LRMs and similar AIs, we leverage what we know about "the best possible" but apply it to these huge, somewhat "black box" solutions.

HOW WELL ARE YOU FOLLOWING THE LECTURE?

What was today's main message about performance?

What are some examples you personally can think of, where performance isn't as simple as picking the right algorithm?

Do you you have the needed background from CS3410?

Why does CS4414 require a lot of coding?

"IDEA MAP" FOR THE WH

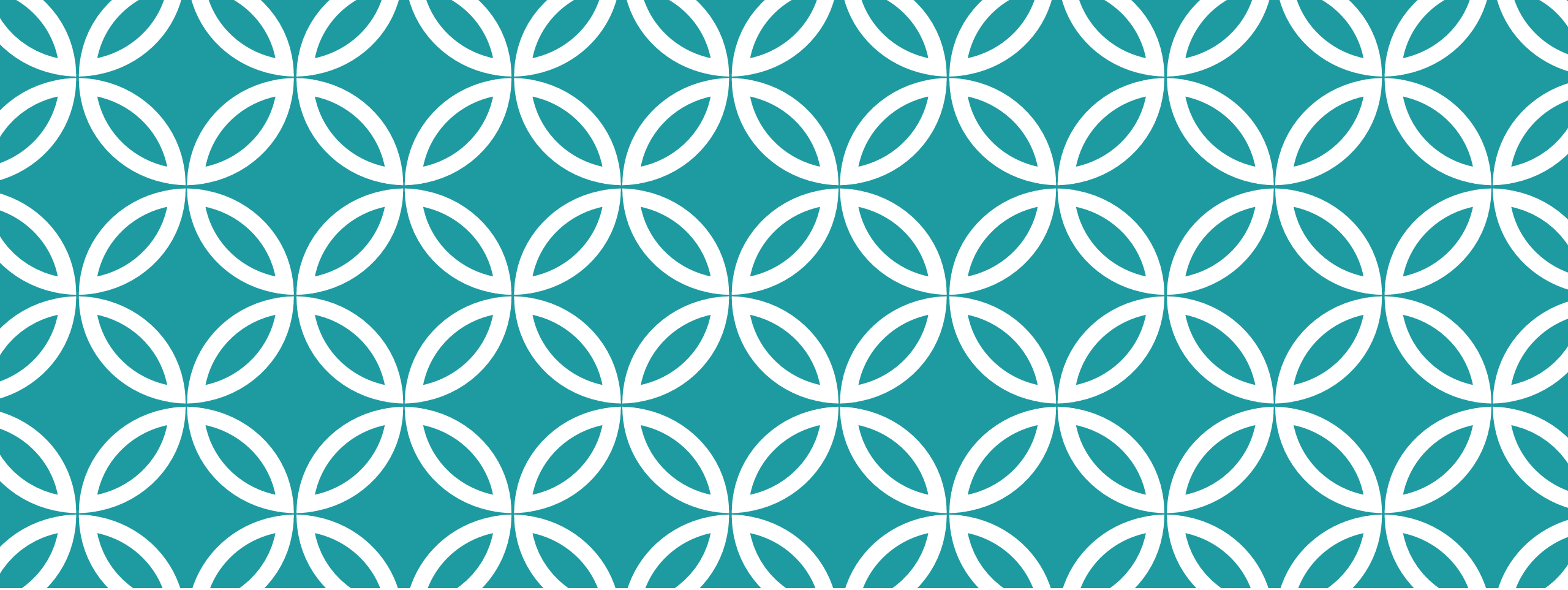
We favor C++ 23 here. The industry seems to favor C++ relative to Rust and other options.

For the MLs, we tend to look at PyTorch over C++ frameworks that often include GPUs.

The application must express your ideas in an elegant, efficient way using a language that promotes correctness and security while mapping cleanly to the hardware
Linux abstractions expose that hardware in easily used forms.

Hardware: Capable of parallel computing, offers a NUMA runtime environment with multiple CPU cores.

Linux: The operating system "manages" the computer for us and translates hardware features into elegant abstractions.



PRACTICAL CONSIDERATIONS

Organizational stuff

LECTURES ARE IN PERSON, NO REMOTE OPTIONS

In past years we posted videos, but in 2025 the course itself evolved sharply and we don't have videos.

Also, for fairness, any form of remote assistance has to be available to everyone.

Our plan? No videos, post slides *a few days after each lecture*.

RECITATIONS ARE REQUIRED & THEY MATTER!

C++ is easy but it takes time and help. And even a person who knows C++ can learn new things from Alicia, Shouxu and Jamal.

Performance debugging tools, however, take time to learn and use, and improving performance is hard!

We teach these skills in recitation, not the main lectures. And the two tracks have different main projects, hence different recitations.

SETUP

We will have everyone using g++ (C++ 23) and the same version of Linux (Ubuntu), and if possible, Visual Studio Code

This makes autograding possible, and also makes TA advising easier.

- For office hours, CS5416 students have their own OH
- Same for Ed Discussions

CANVAS

We use it, a little, but we think of the course website as the main resource for everything

http://www.cs.cornell.edu/courses/cs4414

http://www.cs.cornell.edu/courses/cs5416

These link to the syllabus and to the Ed Discussions site. Final grades will be on CMS and the registrar system, not Canvas

WHY DO WE ALLOW AI HELP?

This isn't a course about coding, it is about performance debugging and performance improvements.

We do start with some coding but rapidly shift to big systems that we didn't implement ourselves.

AIs are tools, like anything else. We should embrace them!

ML FALSE CONFIDENCE

If you play with ML coding, you can generate a great version of examples we will be showing you, even tricky ones.

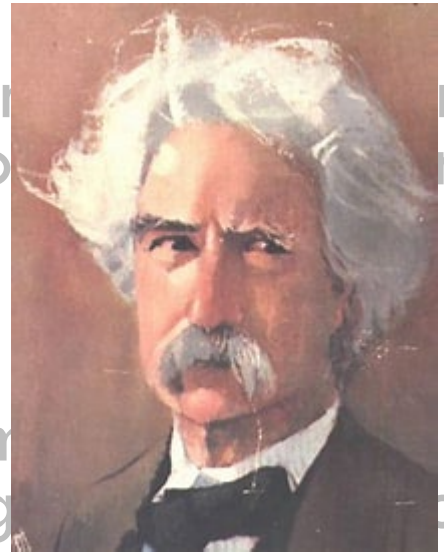
But don't assume that they can solve some random, totally new, weird systems coding problem

- MLs can't generate code for scenarios that don't match the training materials.
- When they seem to work well, this is often because you are in a class settings. MLs train on examples from slides like mine!

COMBATTING "FALSE CONFIDENCE"

"It ain't what you don't know that gets you into trouble; it's what you know for sure that just ain't so."

Generally attributed to Mark Twain (Samuel Clemens).



Lectures include self-study questions to help you think about issues.

COMBATTING "FALSE CONFIDENCE"

Our topic is confusing because there are so many computing elements, so many computing patterns, and the scale of the tasks can be overwhelming.

This isn't an course about things you can memorize – it is about learning techniques that sometimes work well, and figuring out which apply in a given setting

Lectures include self-study questions to help you think about issues.

EXAMS (SOME), HOMEWORK

In CS4414, 50% exams, 50% homework. There will be two prelims but no final. Cornell exam schedules will show the exact dates, times and rooms.

In CS5416, 25% exams, 75% projects. The exams are actually shared with CS4414.

You are required to attend lectures. There are no videos of the 2025 lectures.

CURVE

We curve the course. Two curves: one for 4414, one for 5416

Our feeling is that most students should easily be able to earn a grade in the range from B- to A+. But A+ is "special" and not strictly from a formula. The instructor decides how many to give.

Grades below B- are used if a student really isn't doing well, plus is also skipping lectures.