CSCE 585: Machine Learning Systems

Fall 2024

Pooyan Jamshidi

UNIVERSITY OF South Carolina

A brief self introduction


PhD, DCU, Ireland
2010-2014
Self-adaptation (uncertainty)
Cloud-native & Microservices
RL & Fuzzy control
Visiting researcher,
Google, 2021-2022
Causal AI
Representation Learning
Security of ML Systems
Assistant Prof.,
UofSC, 2018-*
Causal AI Theory & Applications
Robustness of Learned Representations
Multi-Objective BO
[On-device ML, ML Pipelines, (Space) Robotics]
Postdoc1, Imperial
College London, UK,
2014-2016
Bayesian Optimization
Performance Optimization
Big data
Postdoc2, CMU, PA,
2016-2018
Transfer Learning
Robot Learning
White-box Performance Modeling
B.Sc. CS & Math,
Amirkabir, Iran,
1999-2003
Lots of math & theory!
M.Sc., Systems,
Amirkabir, Iran
2003-2006
Expert system for
product design
Software Industry,
2003-2010
Software Engineer
Designer and Architect
Software Project Manager

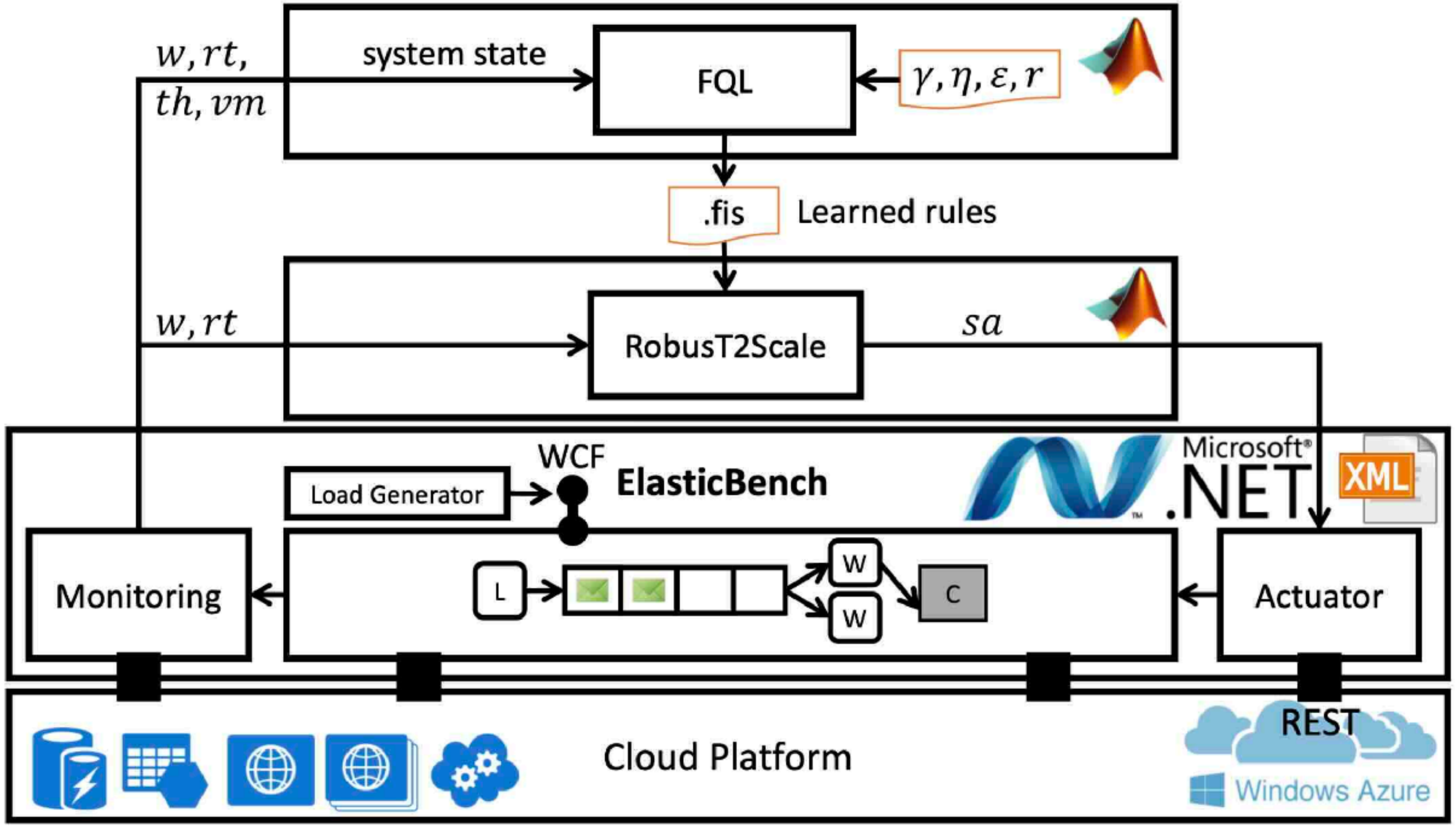
My story …

Machine Learning ➡ Learning Systems

Back in 2010

I started studying the use of Reinforcement Learning and Fuzzy Control for Cloud Auto-scaling.

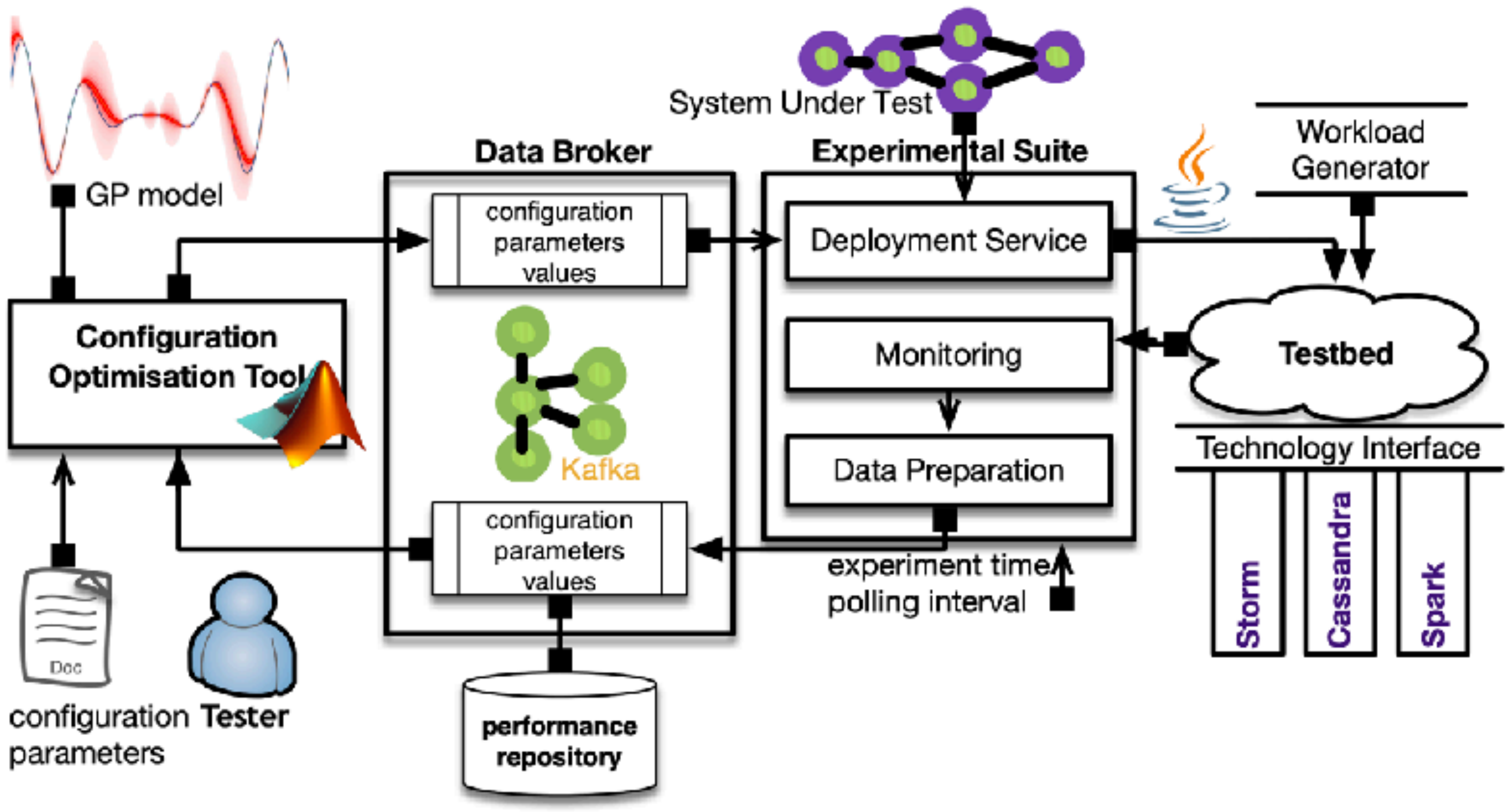
Research was slowed by the speed of RL training … and the fact that system non-functional qualities are difficult to predict

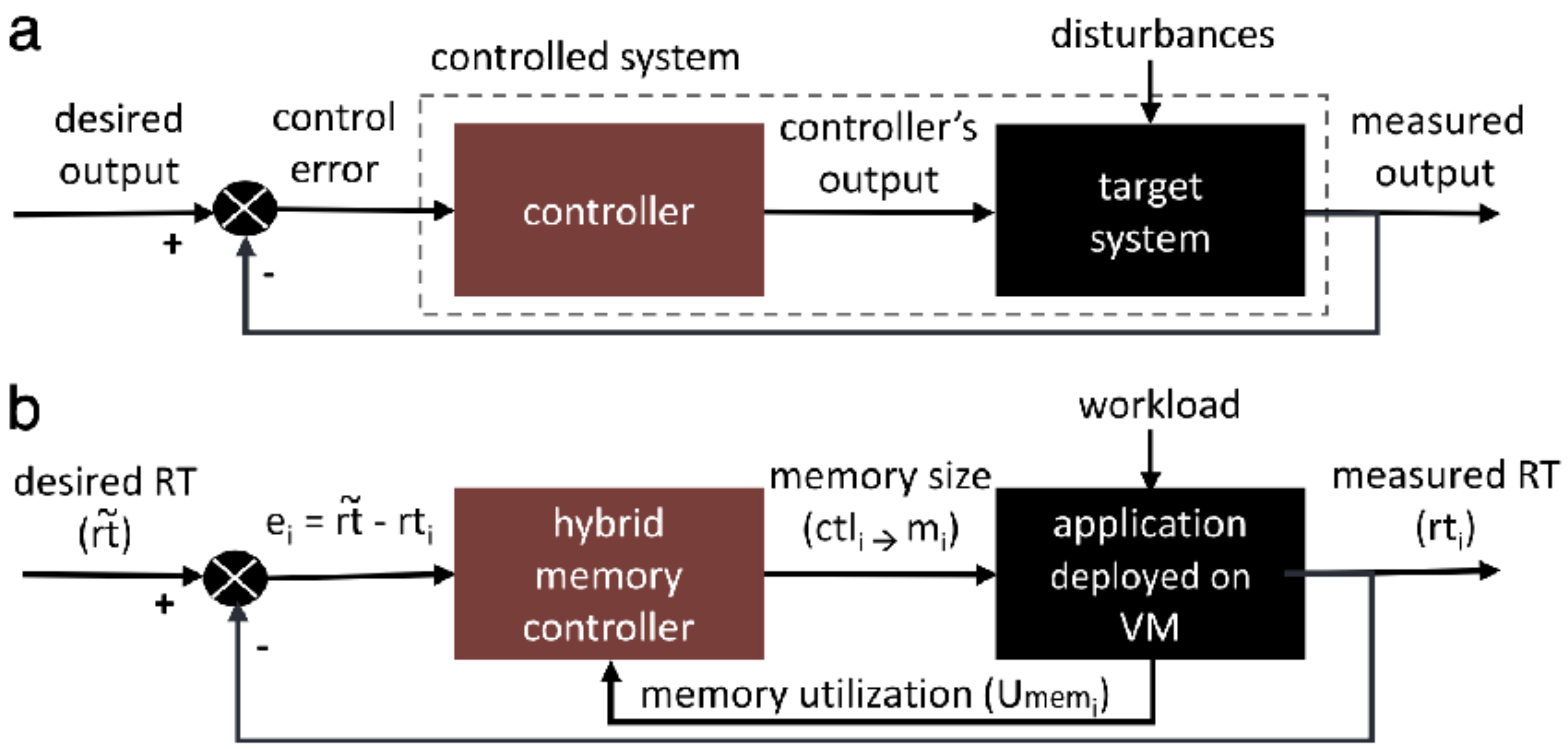


Changed Research Focus

First Postdoc at Imperial College London

I started studying Bayesian Optimization and its applications to Computer Systems (big data, control theory).

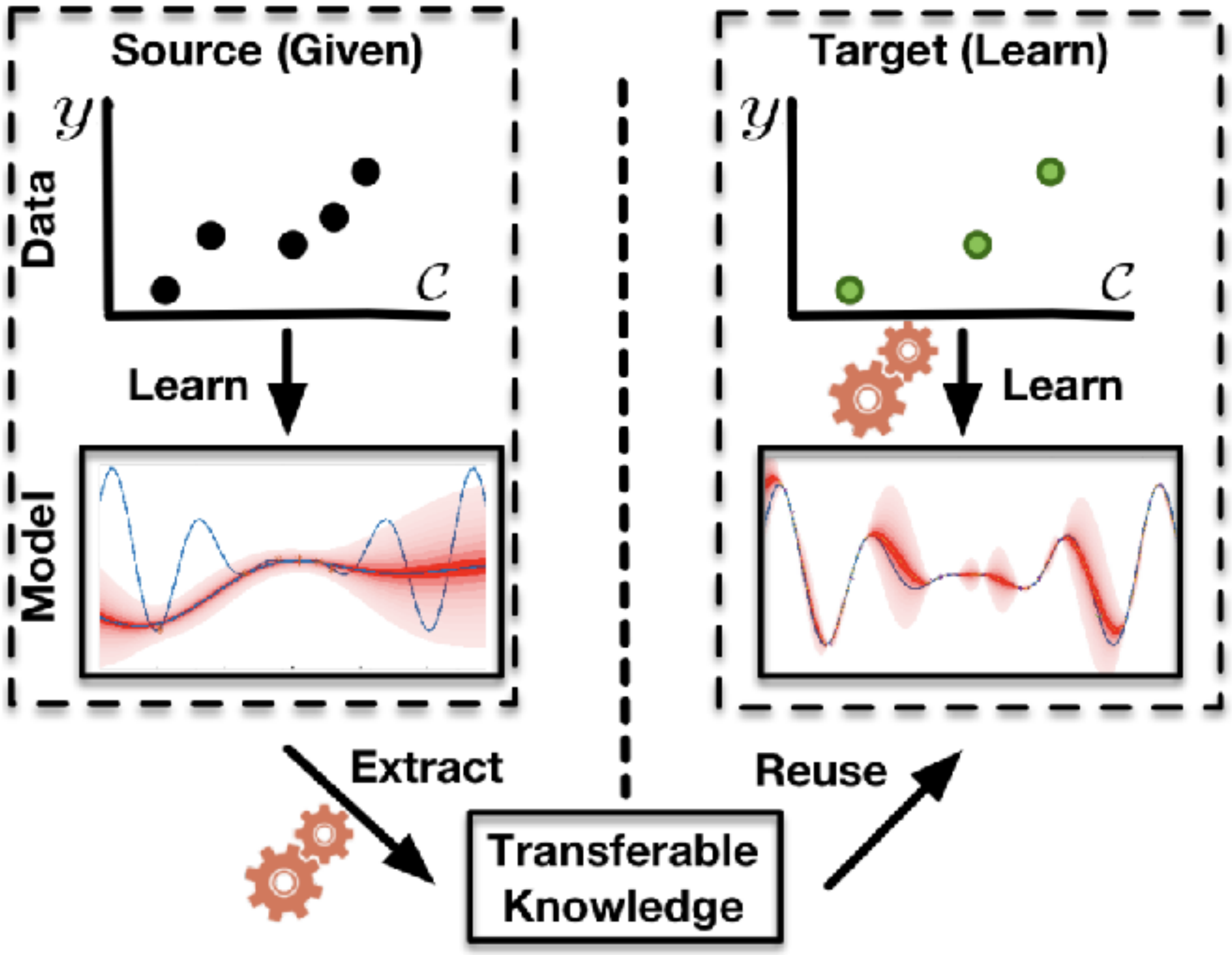


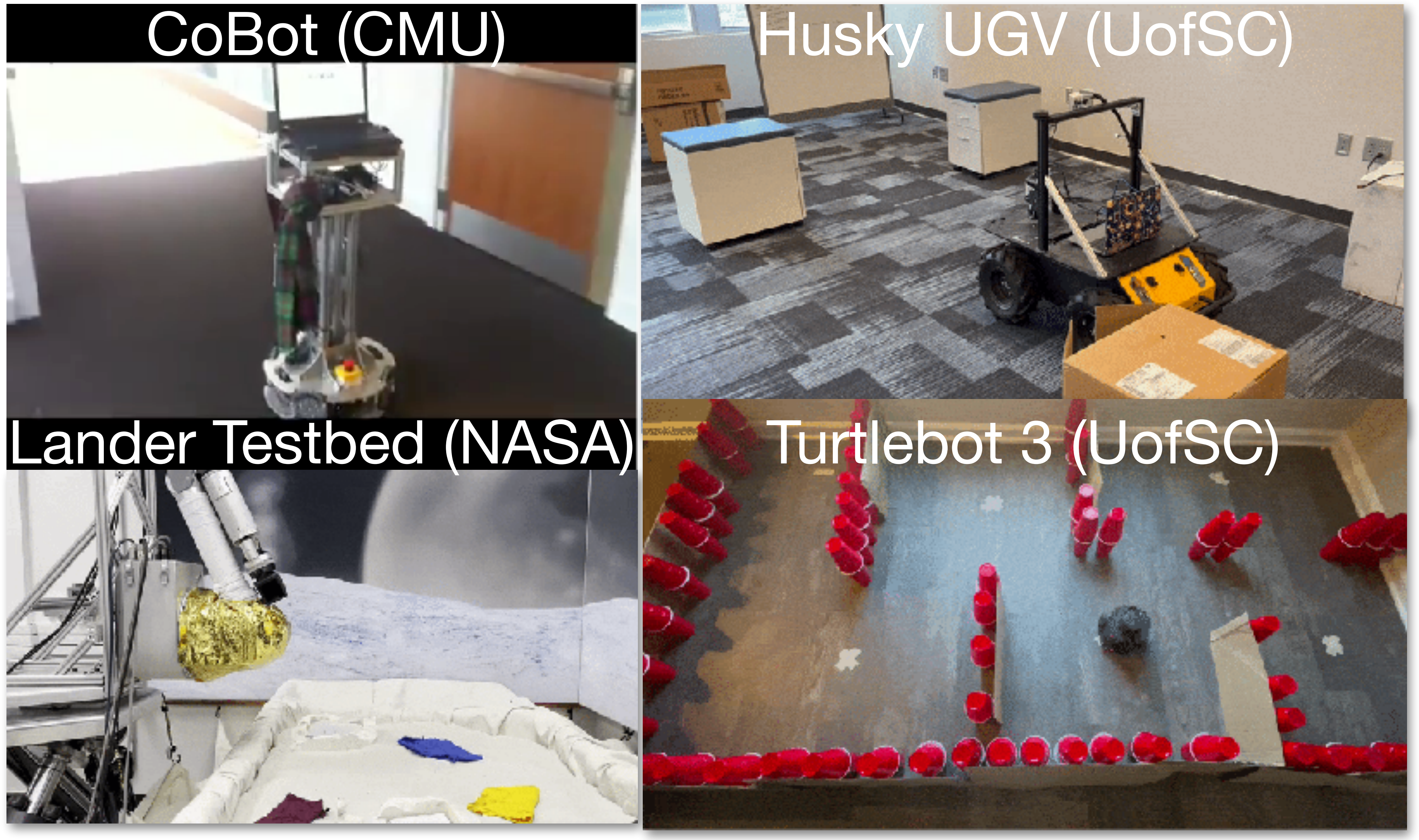


Slight Change in Research Focus

Second Postdoc at Carnegie Mellon University

I started studying Transfer Learning and its applications to Computer Systems (Robots).

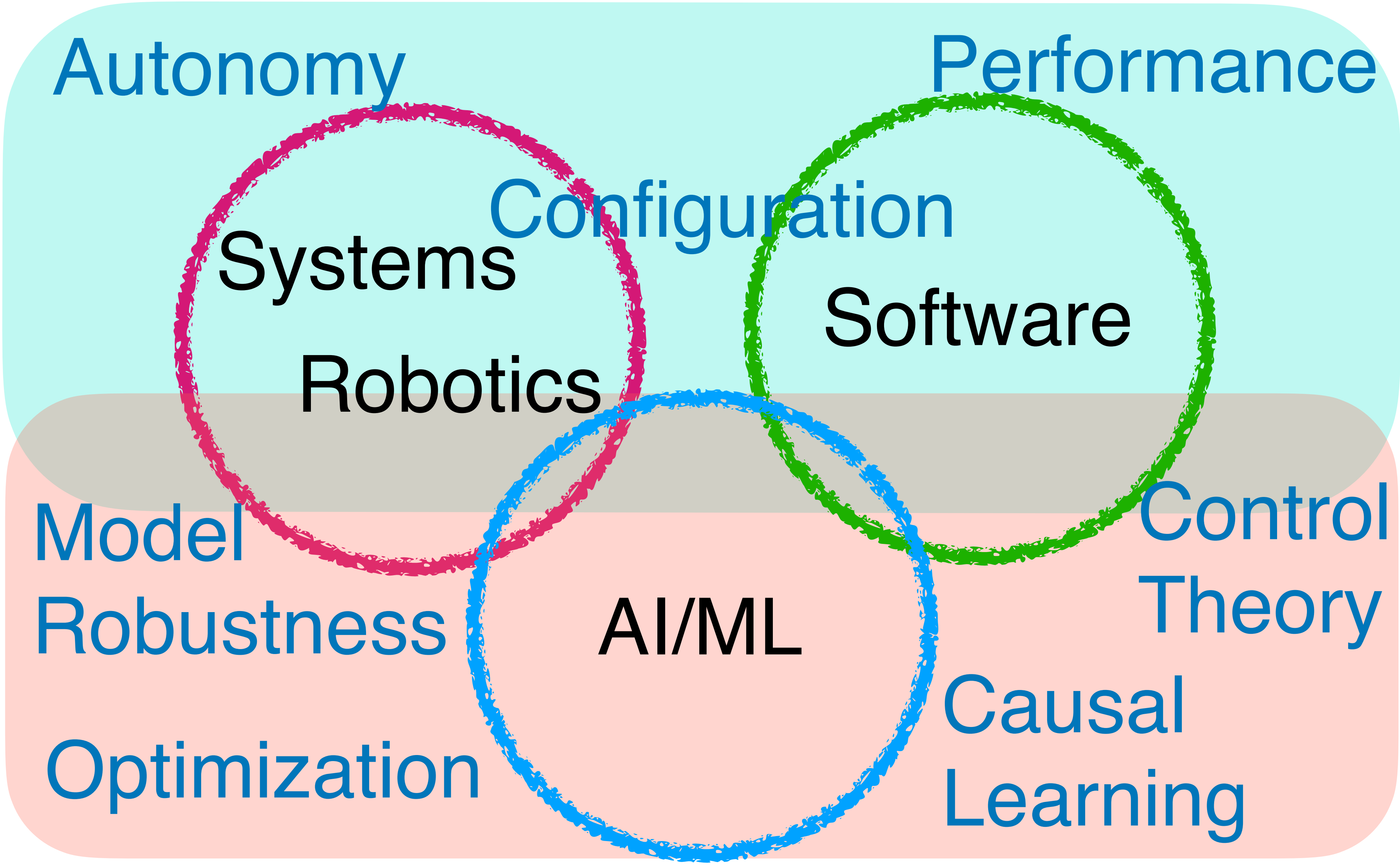




New Research Directions at UofSC

This is my first academic position

I started studying causal AI, adversarial ML, and representation learning and their applications to computer systems (autonomous robots, NASA, and on-device AI).



Artificial Intelligence and Systems Laboratory (AISys)

https://pooyanjamshidi.github.io/AISys/

Research Areas:
- Causal AI
- ML for Systems
- Systems for ML
- Adversarial ML
- Robot Learning
- Representation Learning


Fatemeh Ghofrani
(PhD student)


Abir Hossen
(PhD student)


Sonam Kharde
(Postdoc)


Samuel Whidden
(Undergraduate)


Rasool Sharifi
(PhD student)


Saeid Ghafouri
(PhD student)


Hamed Damirchi
(PhD student)


Shahriar Iqbal
(PhD student)


Mehdi Yaghouti
(Postdoc)


Kimia Noorbakhsh
(Undergraduate)

Sponsors:
Google
NSF NASA DARPA

Collaborators:

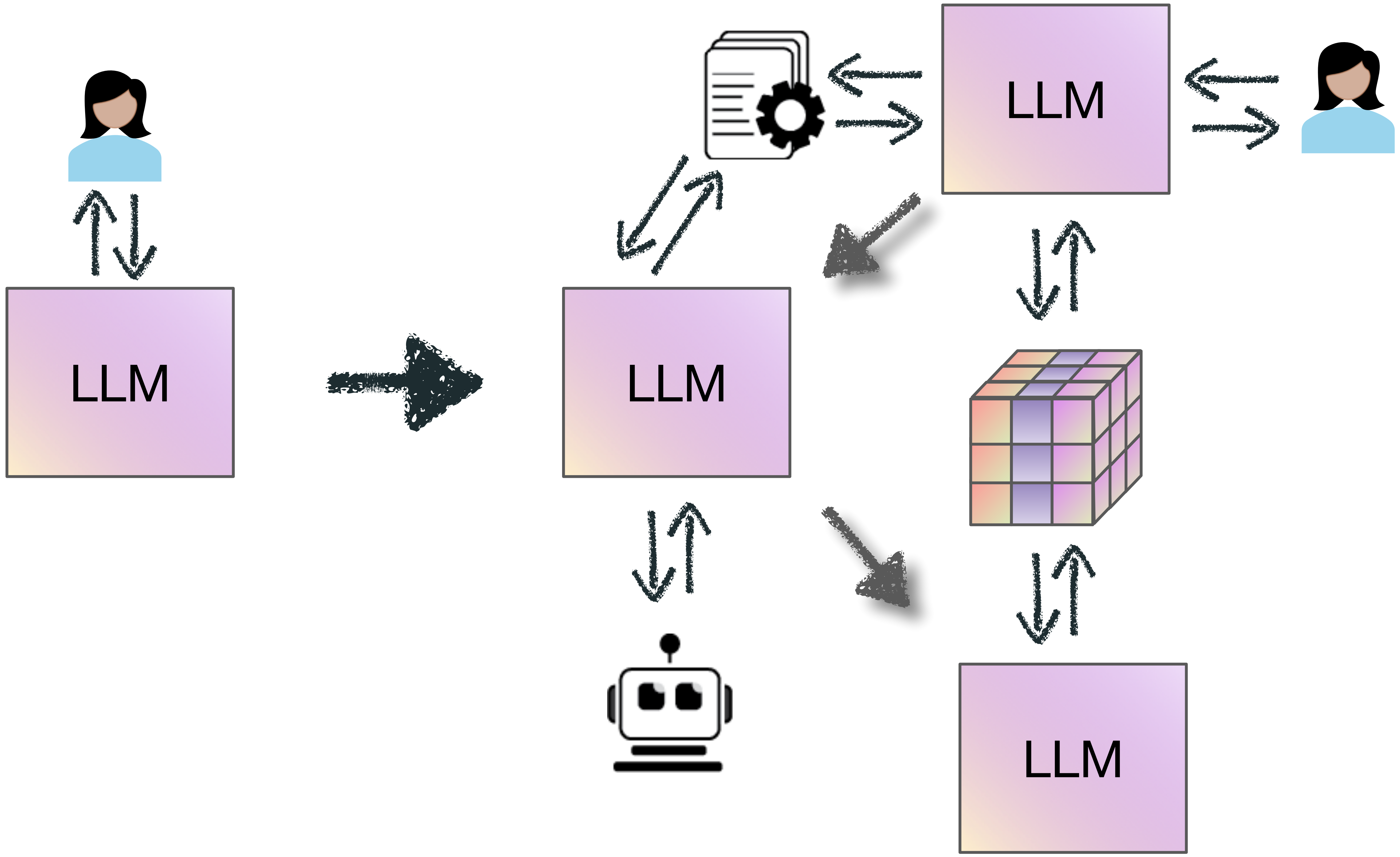
What I am interested in these days the most…

And it is very relevant to this course!

Modularity &

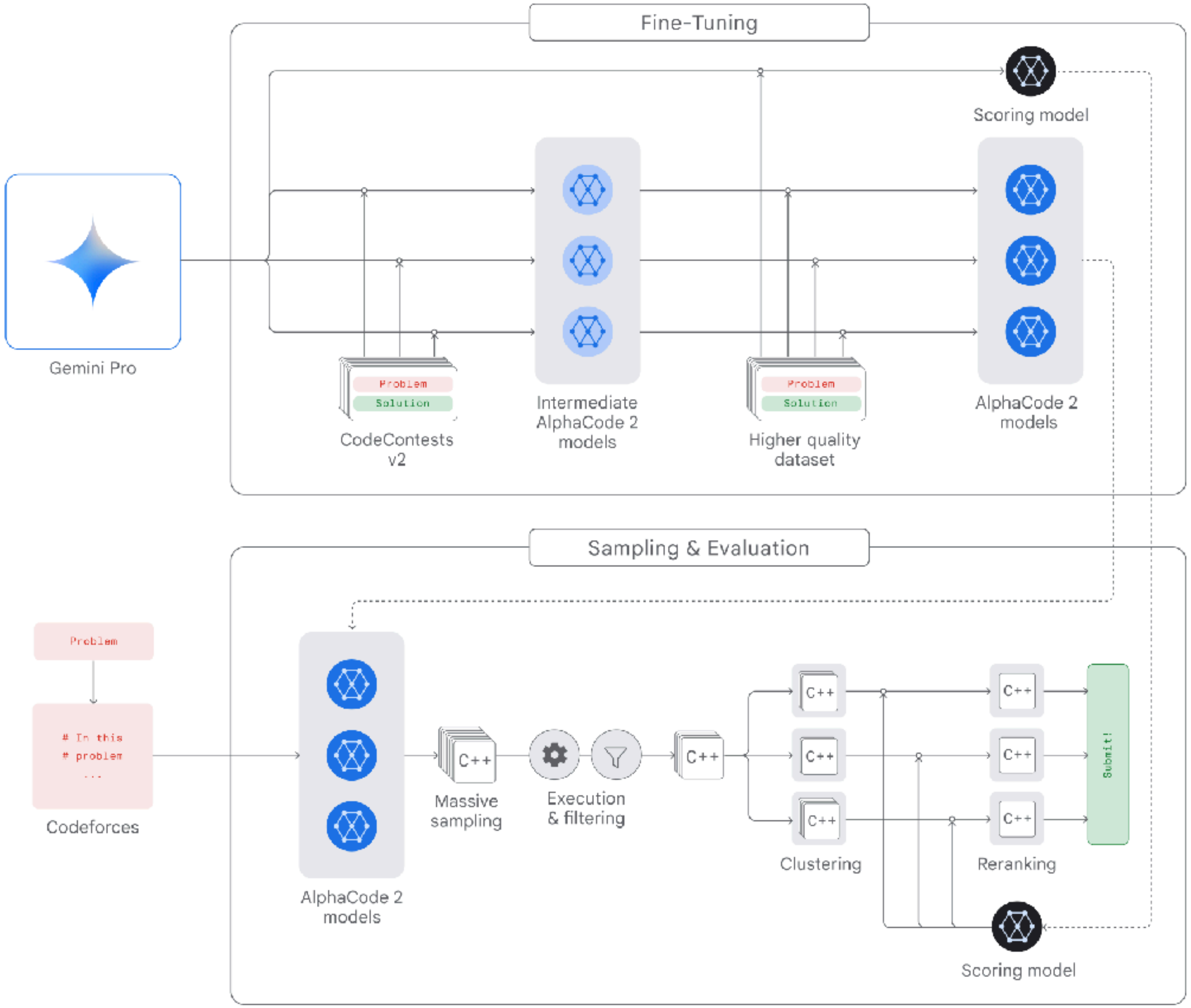
Multi-Component **Composed (ML) Systems**!

State-of-the-art AI results are increasingly obtained by systems composed of multiple components, not just monolithic models



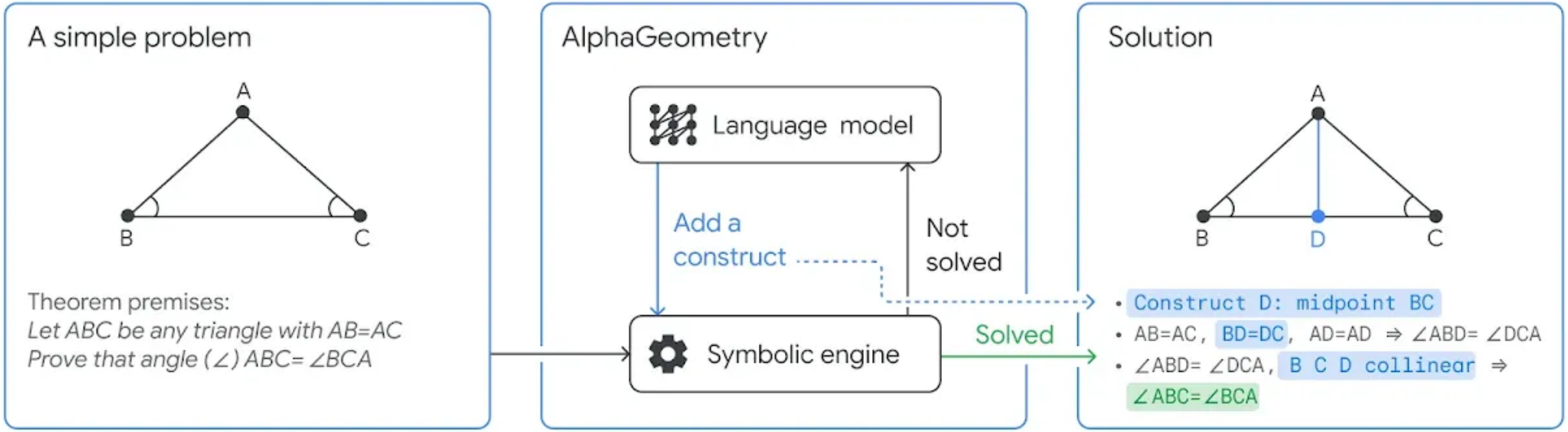
State-of-the-art AI results are increasingly obtained by systems composed of multiple components, not just monolithic models

Google's AlphaCode 2 set state-of-the-art results in programming through a carefully engineered system that uses LLMs to generate up to 1 million possible solutions for a task and then filter down the set.



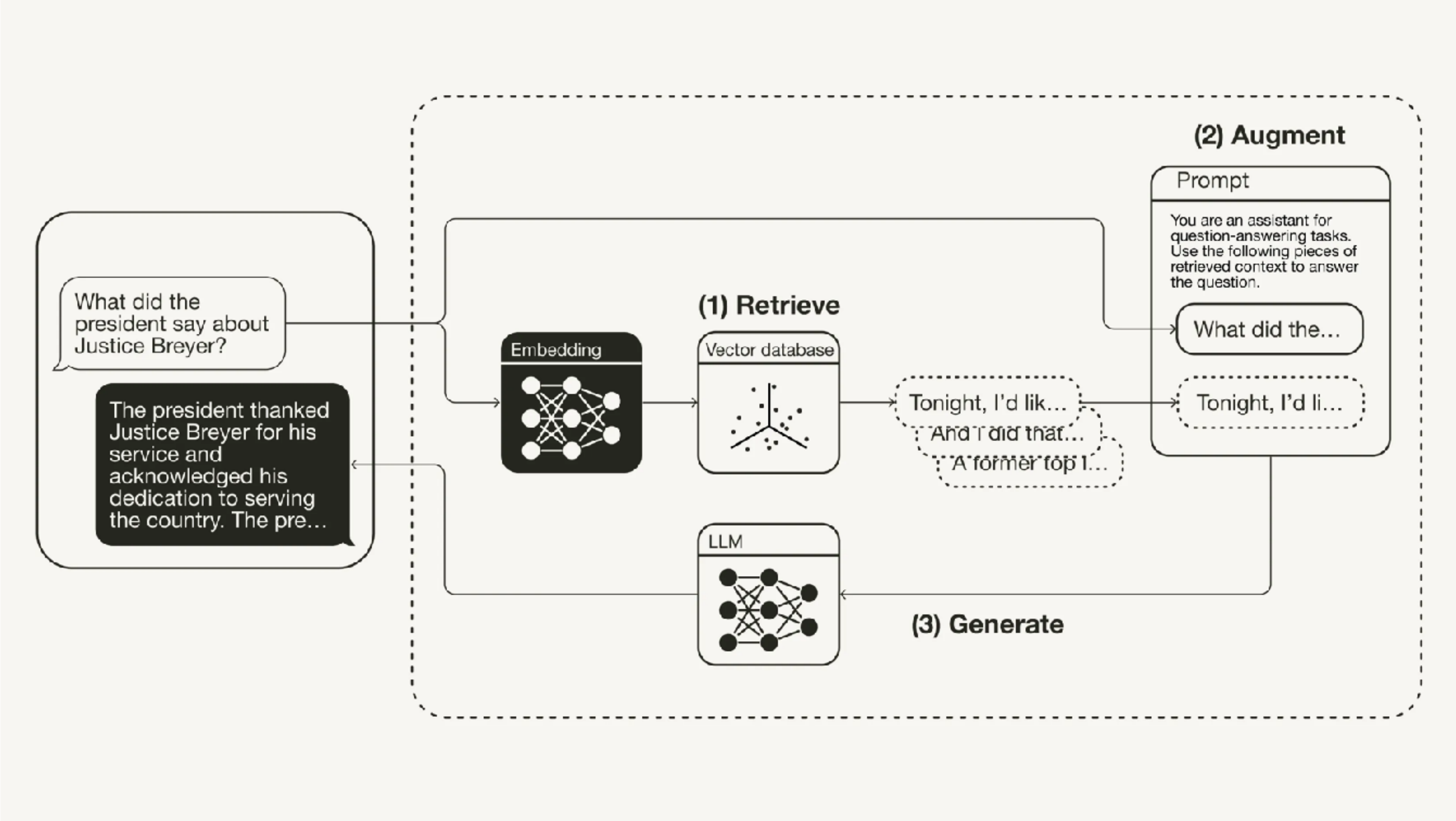
State-of-the-art AI results are increasingly obtained by systems composed of multiple components, not just monolithic models

AlphaGeometry combines an LLM with a traditional symbolic solver to tackle Olympiad problems.



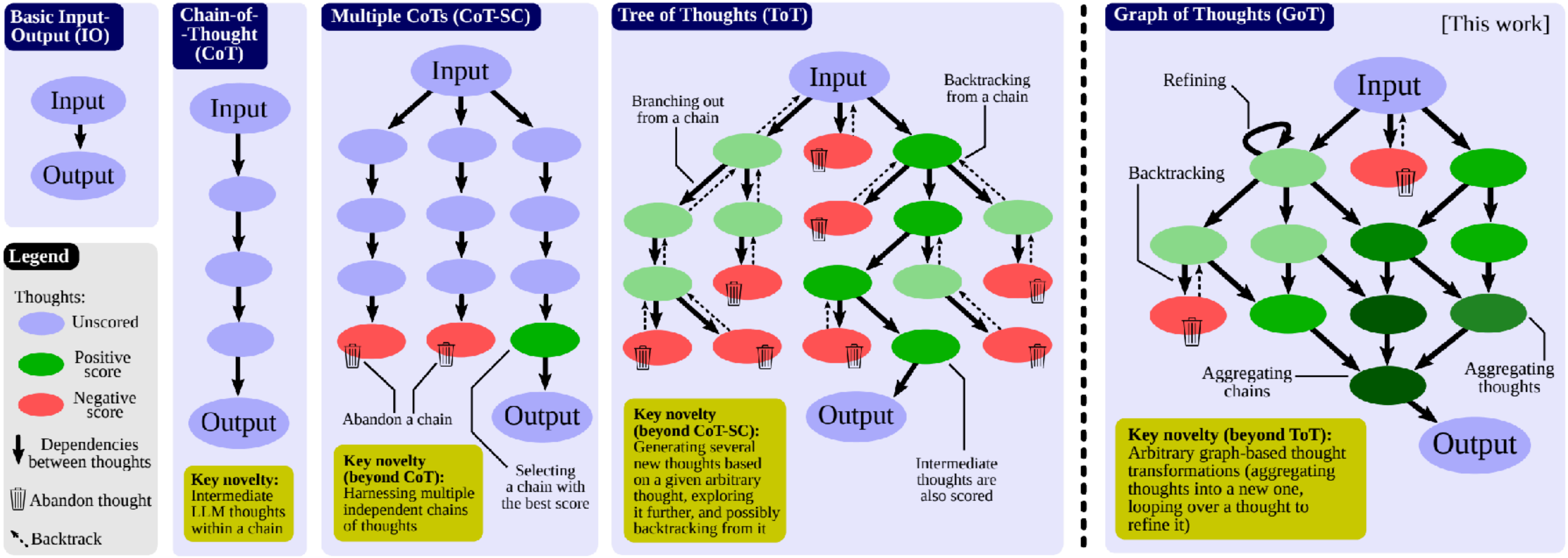
State-of-the-art AI results are increasingly obtained by systems composed of multiple components, not just monolithic models

~60% of LLM applications use some form of **retrieval-augmented generation (RAG)**



State-of-the-art AI results are increasingly obtained by systems composed of multiple components, not just monolithic models

…and 30% use **multi-step chains**.



State-of-the-art AI results are increasingly obtained by systems composed of multiple components, not just monolithic models

Github Copilot uses carefully tuned smaller models and various search heuristics to provide results.

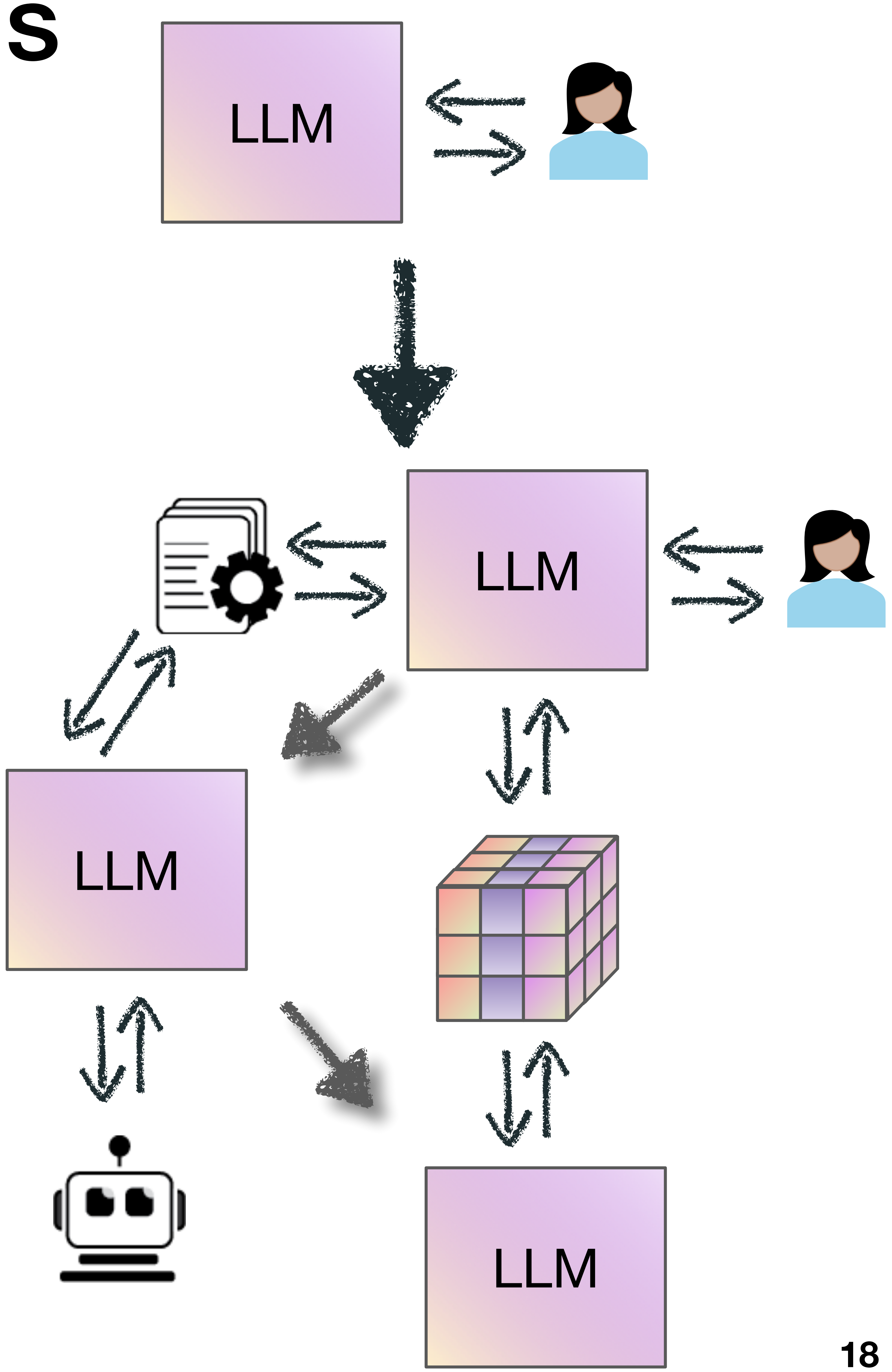
State-of-the-art AI results are increasingly obtained by systems composed of multiple components, not just monolithic models

Google's Gemini launch post measured its MMLU (Massive Multitask Language Understanding) benchmark results using a new CoT@32 inference strategy that calls the model 32 times.

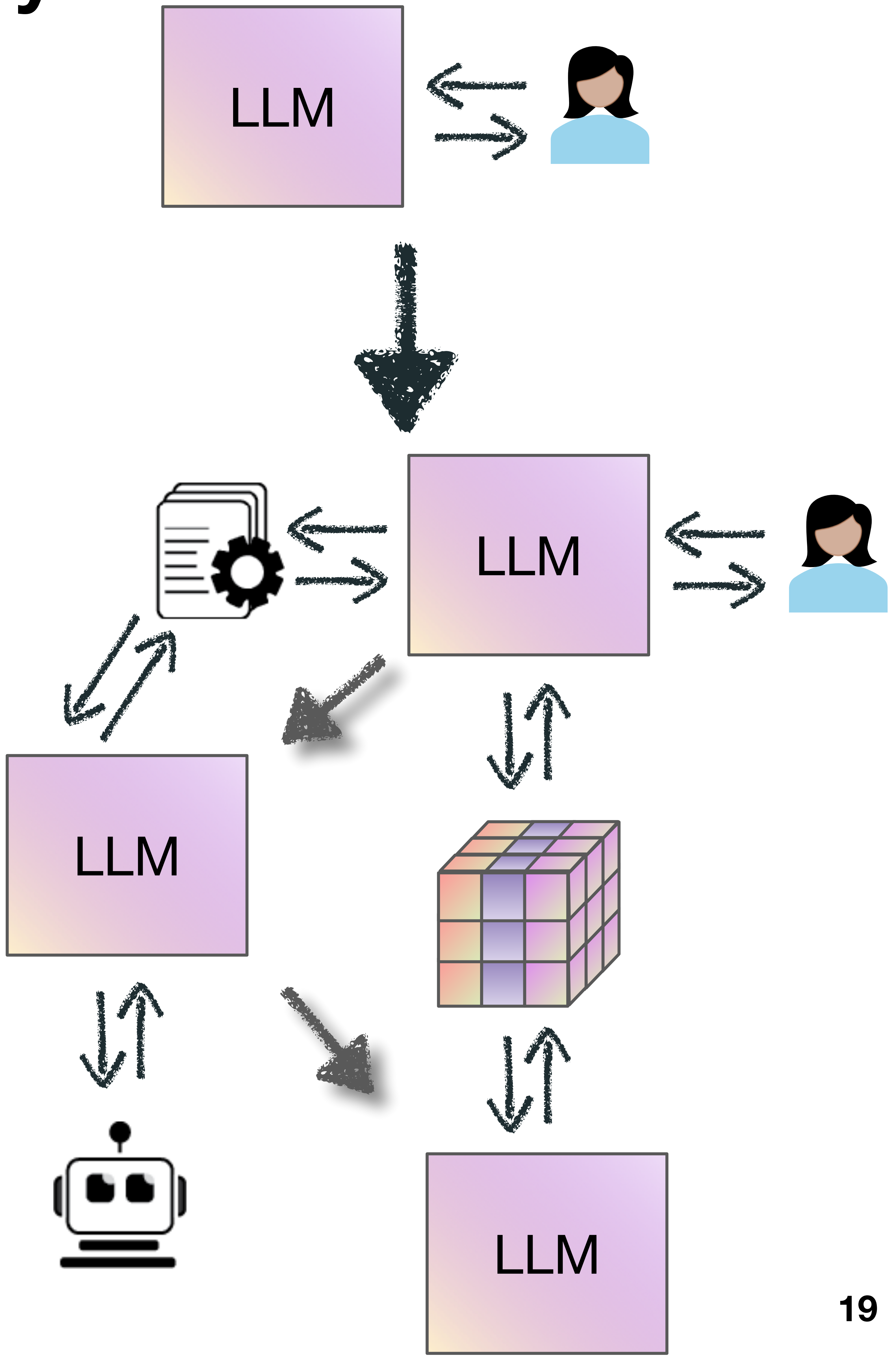
	Gemini Ultra	Gemini Pro	GPT-4	GPT-3.5	PaLM 2-L	Claude 2	Inflect-ion-2	Grok 1	LLAMA-2
MMLU Multiple-choice questions in 57 subjects (professional & academic) (Hendrycks et al., 2021a)	**90.04%** CoT@32* 83.7% 5-shot	79.13% CoT@8* 71.8% 5-shot	87.29% CoT@32 (via API**) 86.4% 5-shot (reported)	70% 5-shot	78.4% 5-shot	78.5% 5-shot CoT	79.6% 5-shot	73.0% 5-shot	68.0%***

The paradigm shift from monolithic to modular-composed machine learning systems

- **Modular-composed ML Systems** are a class of modern computer systems that tackle AI/ML tasks using:
  - Multiple **interacting** and **interdependent** **components**,
  - including multiple calls to **models**, **search** & **retrieval** algorithms, and external **tools**.
- In contrast, **Monolithic ML Systems** are simply traditional ML Systems that call a **statistical model** at the backend.
  - e.g., a **Transformer** that predicts the next token in text.



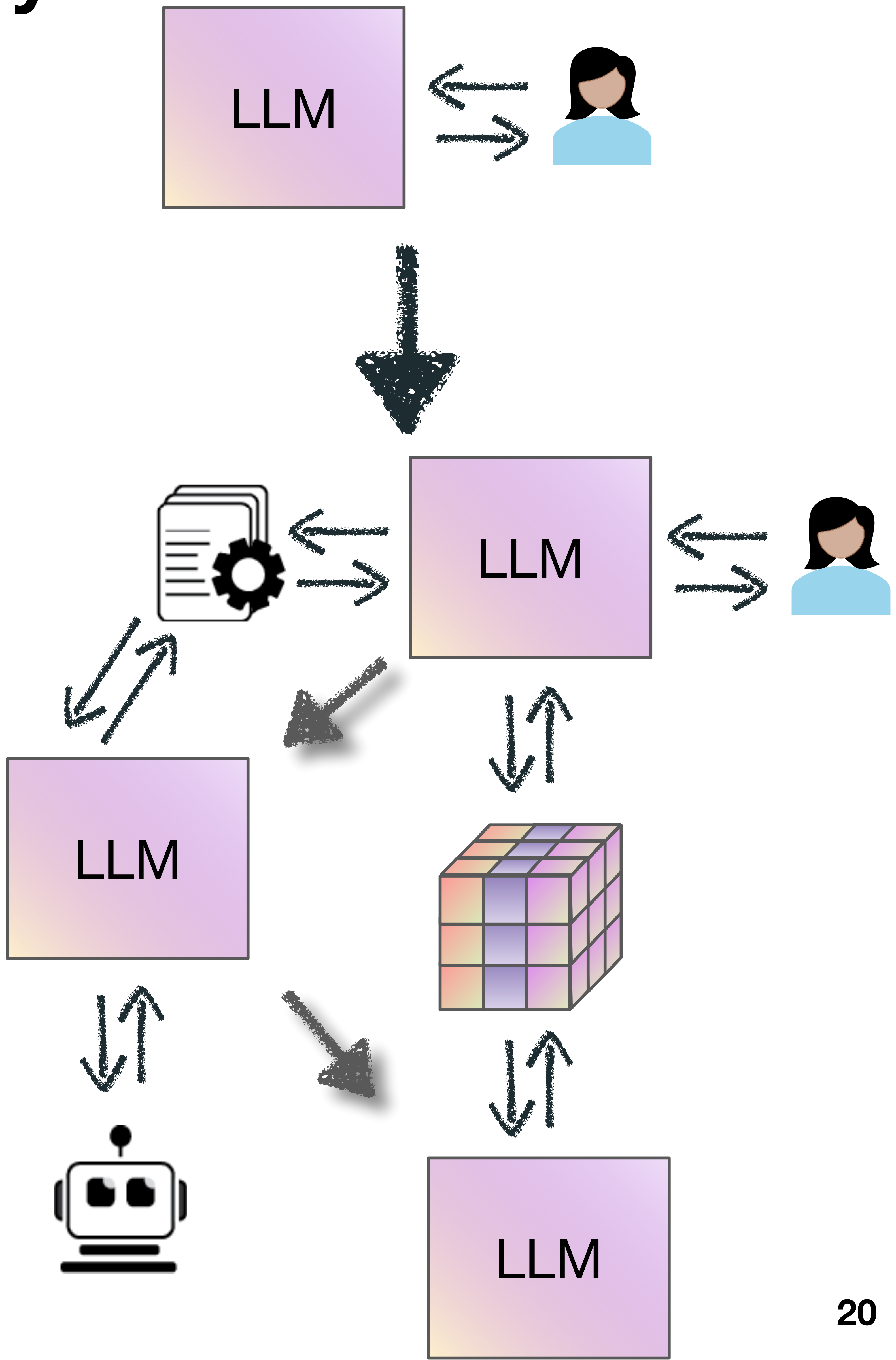
This paradigm shift to modular-composed ML systems opens up new opportunities for computer systems research



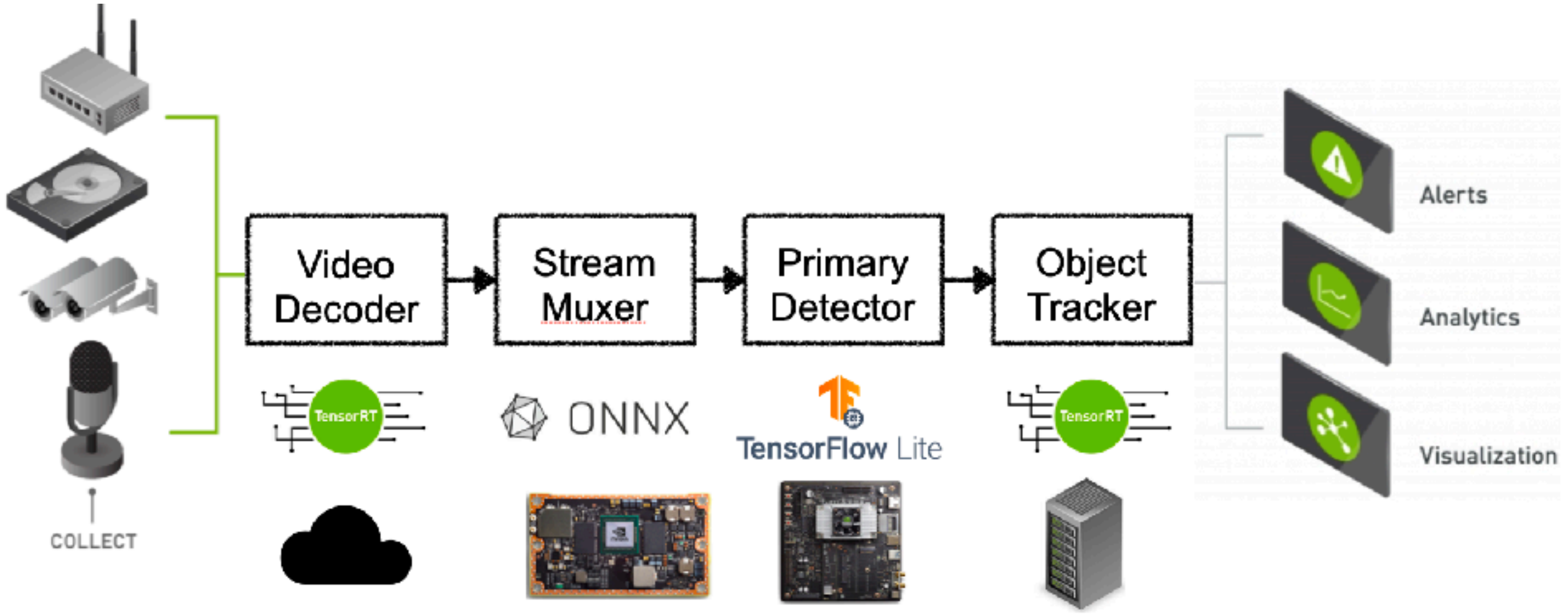
- Design space exploration
  - With an SLA of 100 milliseconds for RAG, should we budget to spend 20 ms on the retriever and 80 on the LLM, or the other way around?
- Performance tradeoff and optimization
  - Modular-composed systems contain non-differentiable components.
  - Performance optimization for pipelines of pretrained LLMs and other components.

This paradigm shift to modular-composed ML systems opens up new opportunities for computer systems research

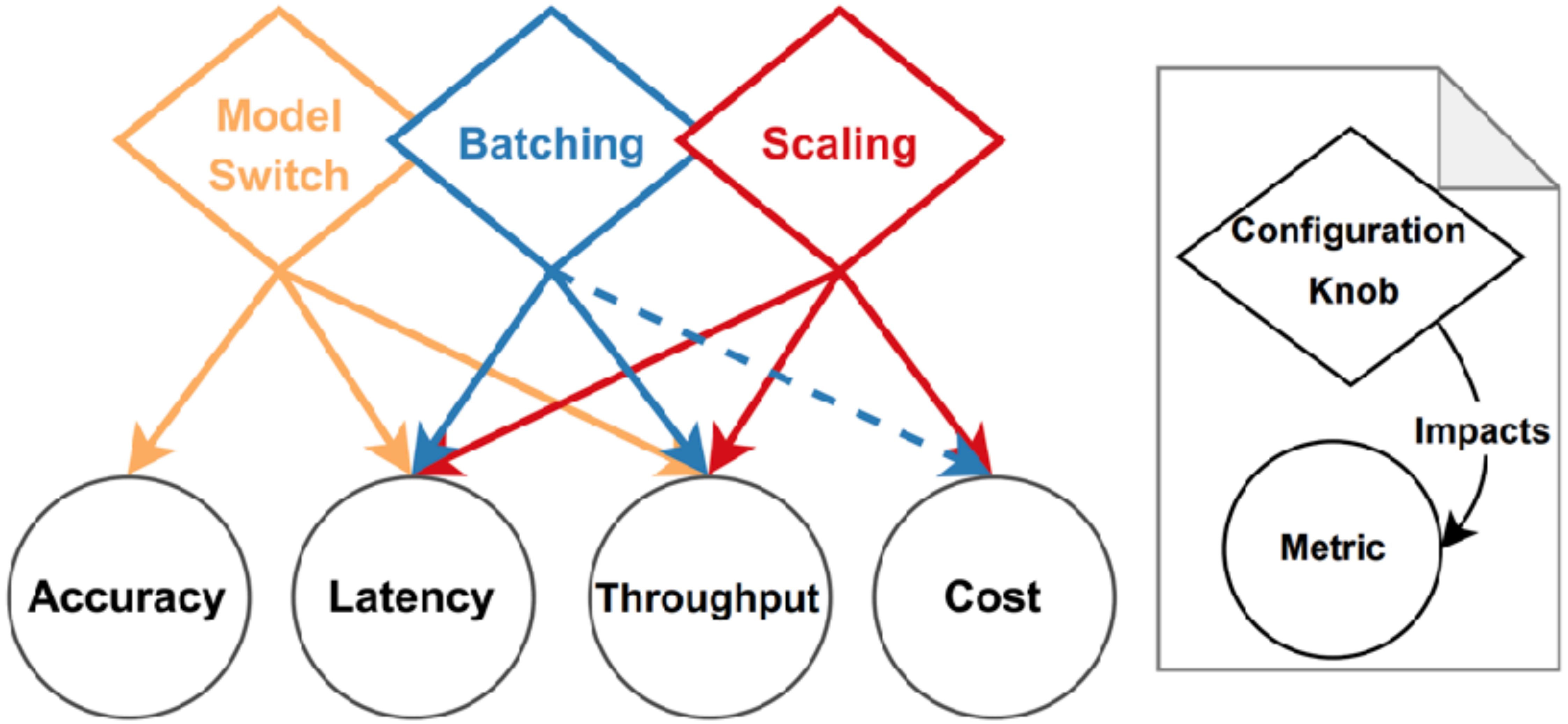
- This shift to modular-composed systems opens many **interesting systems questions.**
- It is also exciting because it means leading AI results can be achieved through clever **systems ideas**, not just scaling up training.



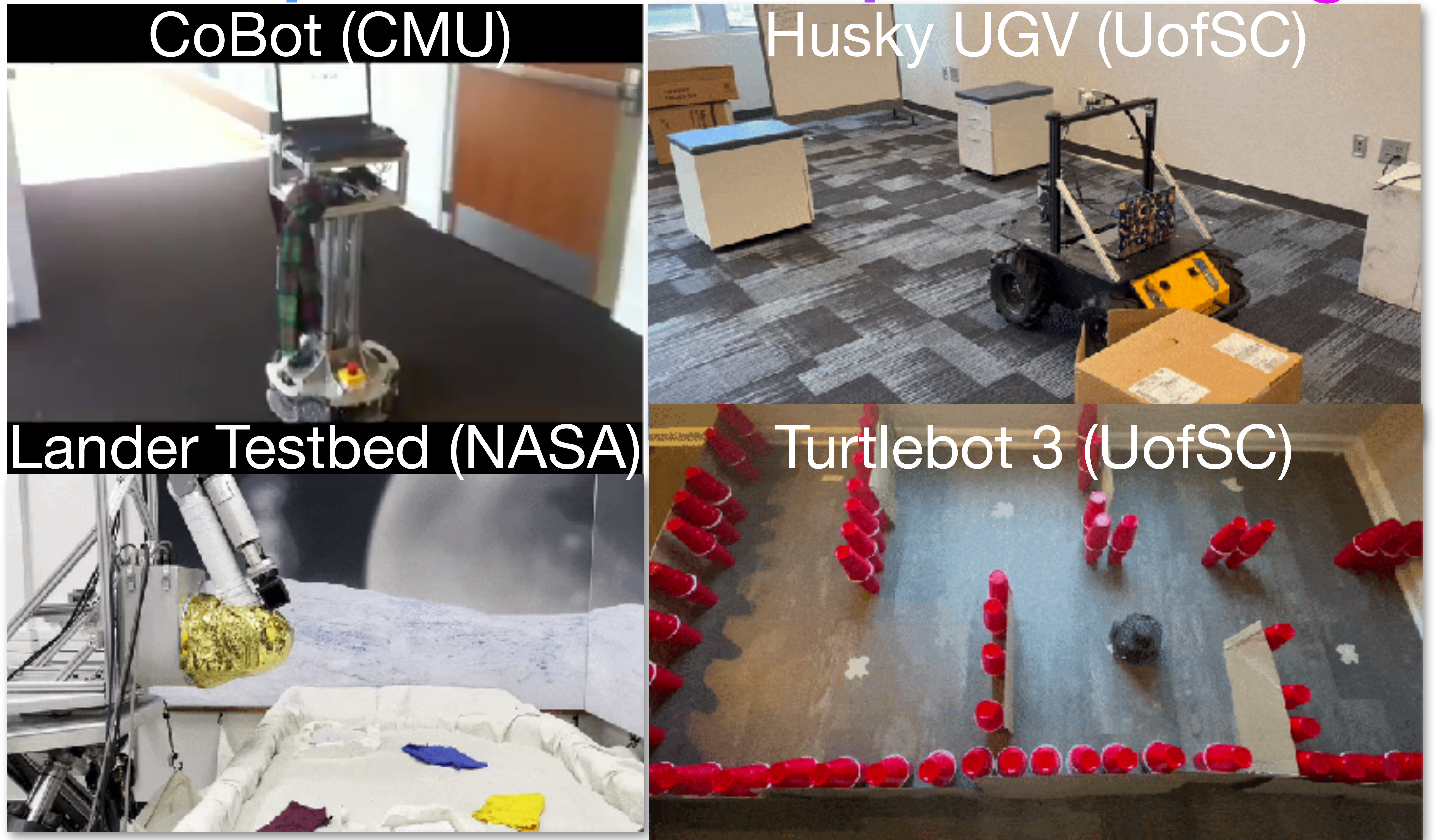
The variability space (design space) of (composed) systems is exponentially increasing



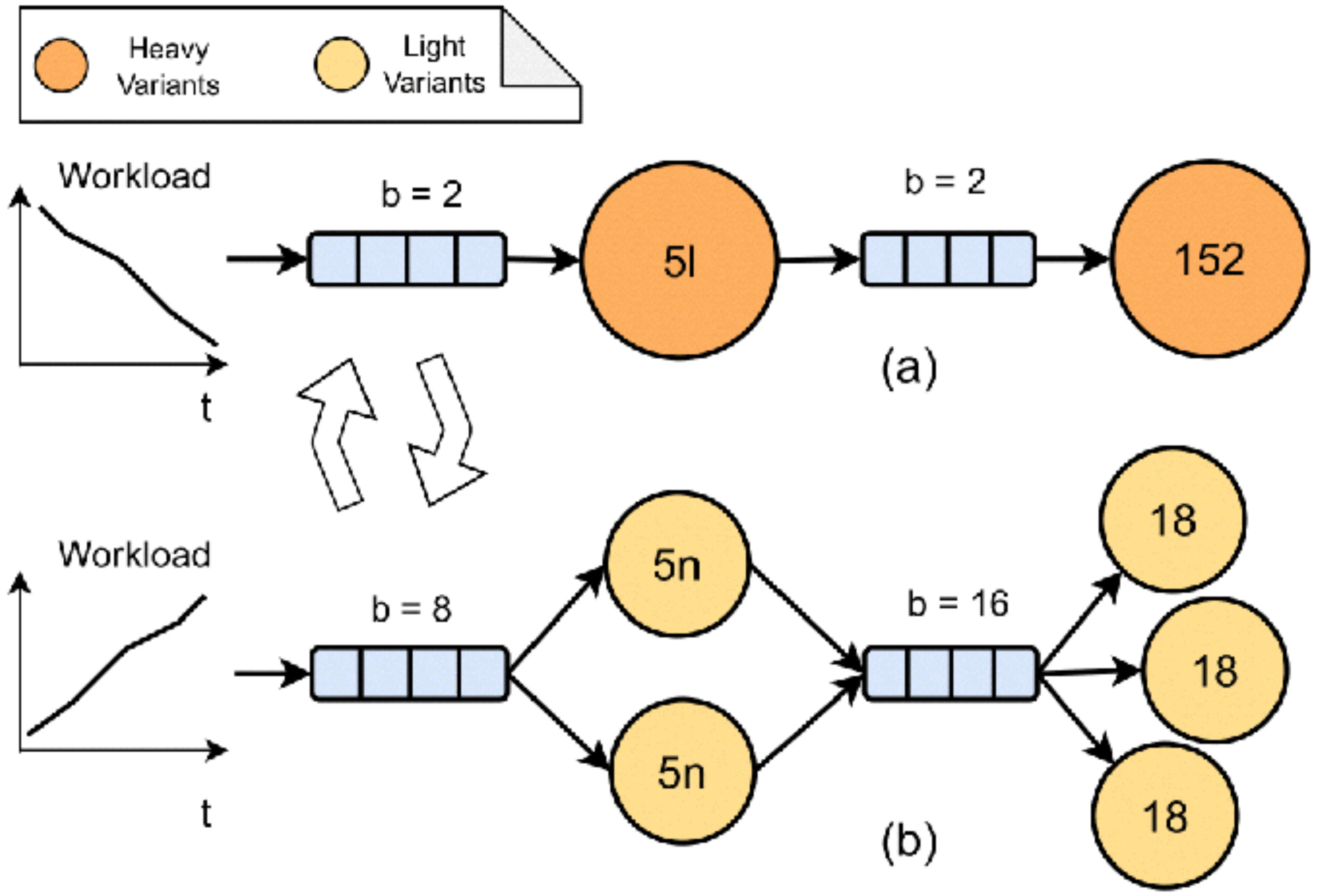
Performance goals are competing and users have preferences over these goals



Systems operate in uncertain environments with imperfect and incomplete knowledge

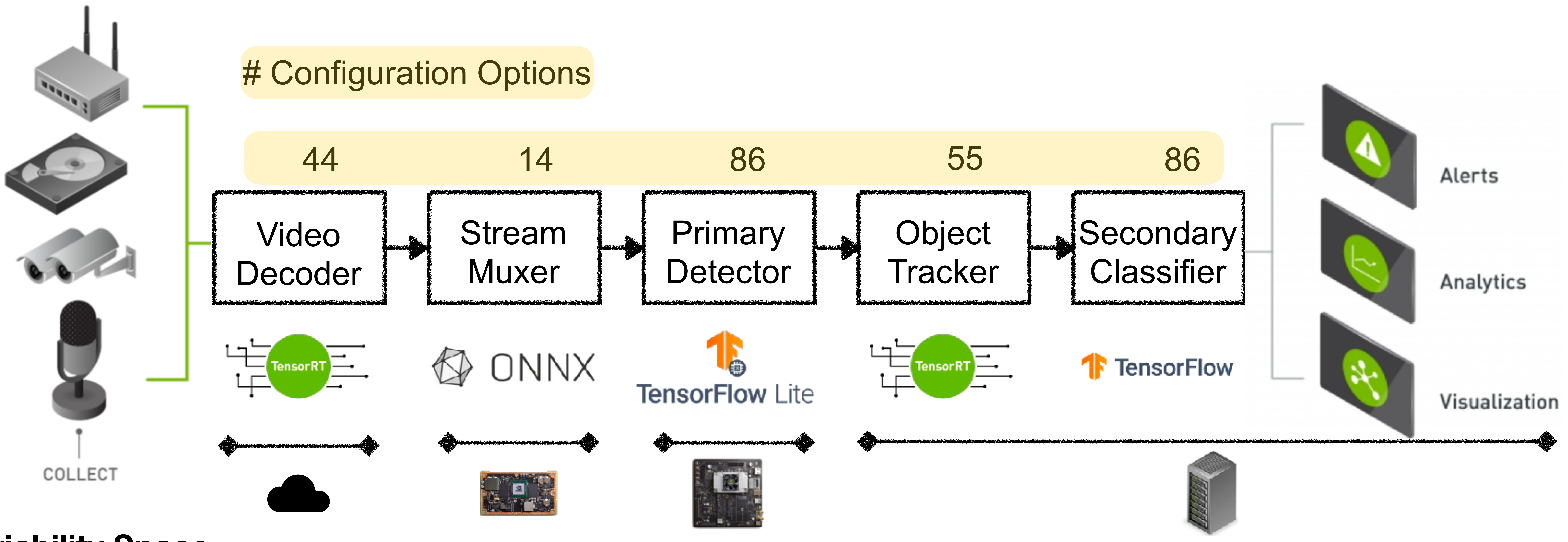


Goal: Enabling users to find the right quality tradeoff



The variability space of today's systems is exponentially increasing

Systems are heterogeneous, multiscale, multi-modal, and multi-stream

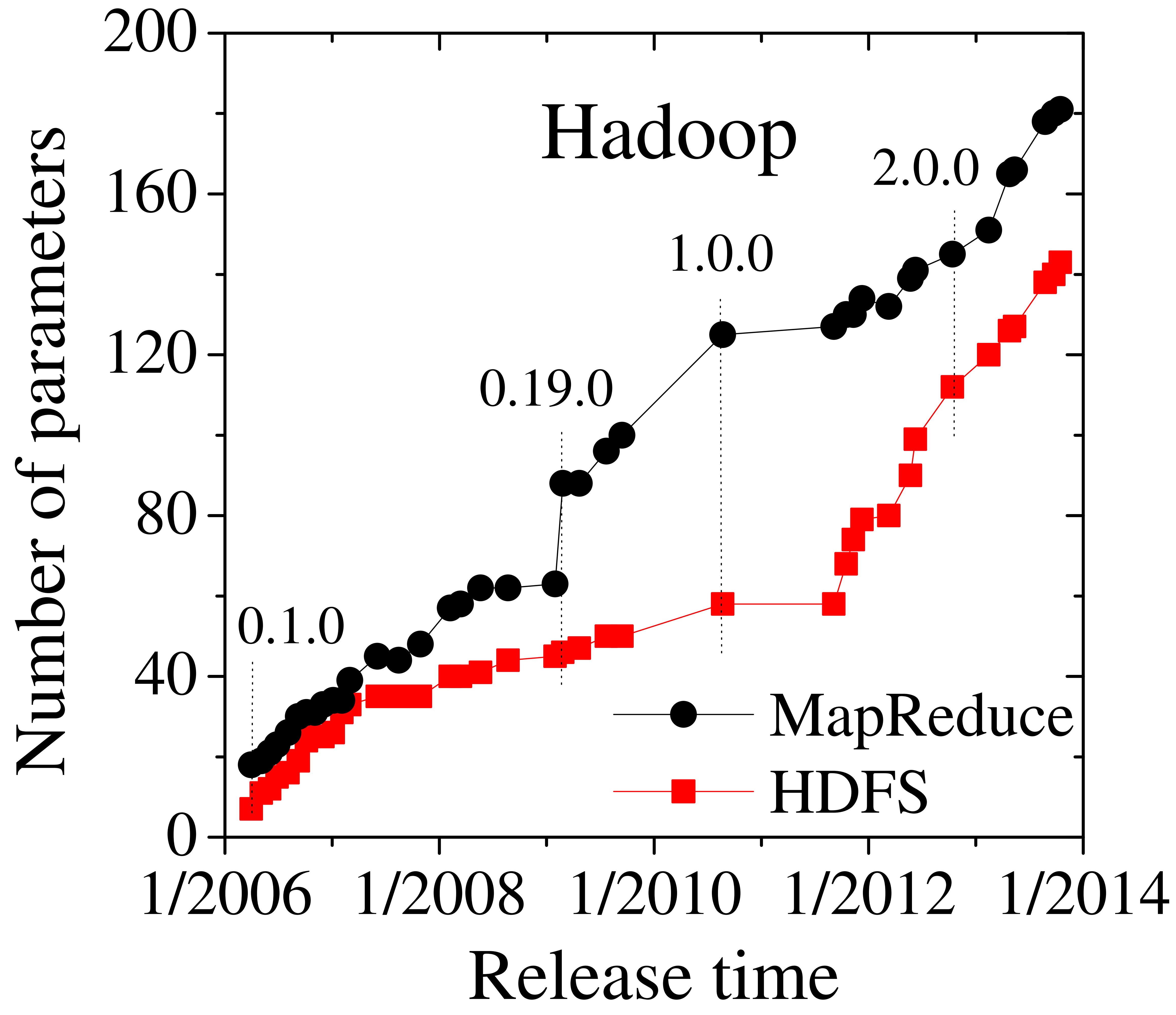


Variability Space =
Algorithm Selection +
Configuration Space +
System Architecture +
Deployment Environment


More configurations than estimated
atoms in the universe

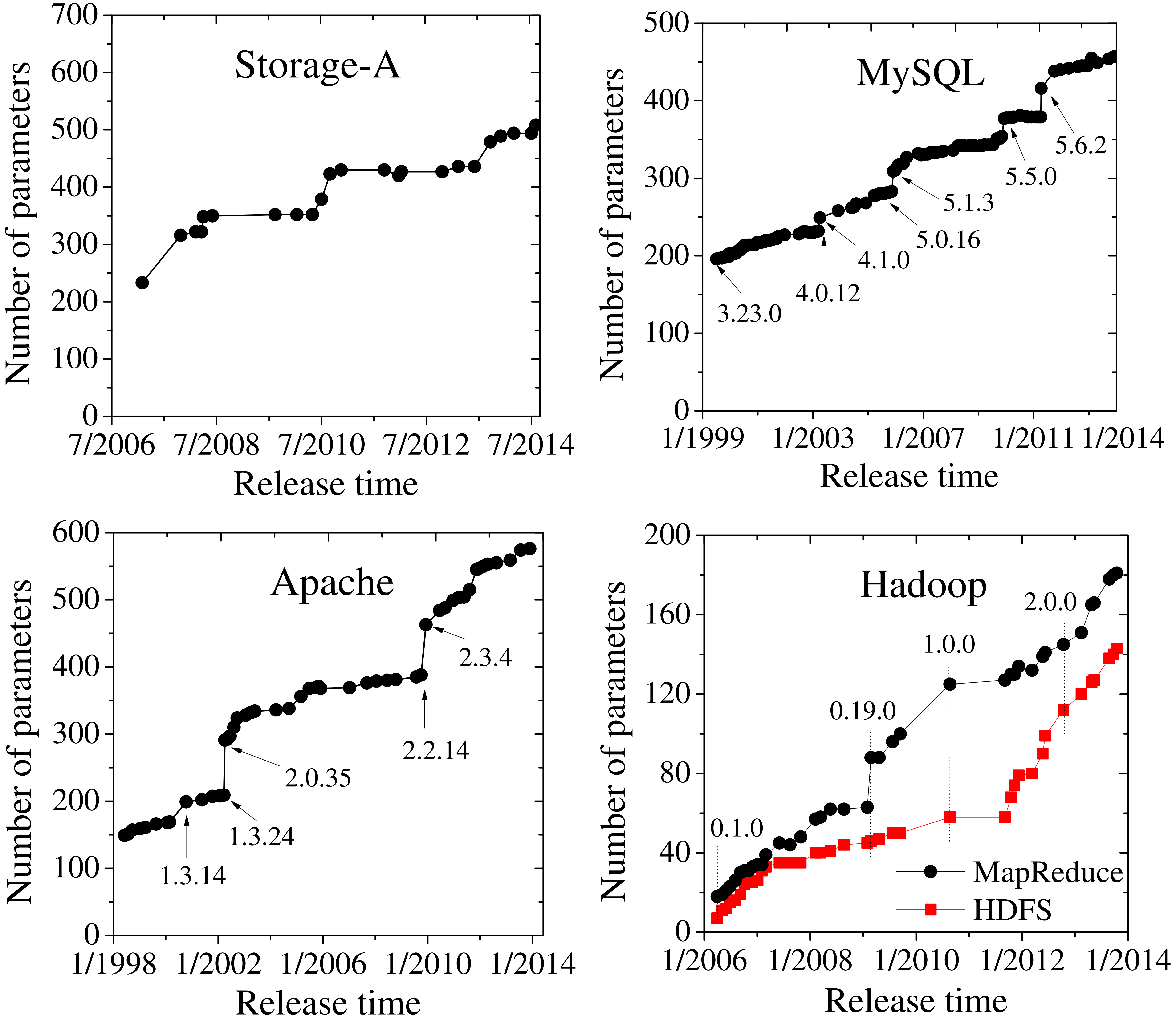
```yaml

drpc.port: 3772
drpc.worker.threads: 64
drpc.max_buffer_size: 1048576
drpc.queue.size: 128
drpc.invocations.port: 3773
drpc.invocations.threads: 64
drpc.request.timeout.secs: 600
drpc.childopts: "-Xmx768m"
drpc.http.port: 3774
drpc.https.port: -1
drpc.https.keystore.password: ""
drpc.https.keystore.type: "JKS"
drpc.http.creds.plugin: org.apache.storm.security.auth.DefaultHttpCredentialsPlugi
drpc.authorizer.acl.filename: "drpc-auth-acl.yaml"
drpc.authorizer.acl.strict: false

transactional.zookeeper.root: "/transactional"
transactional.zookeeper.servers: null
transactional.zookeeper.port: null

## blobstore configs
supervisor.blobstore.class: "org.apache.storm.blobstore.NimbusBlobStore"
supervisor.blobstore.download.thread.count: 5
supervisor.blobstore.download.max_retries: 3
supervisor.localizer.cache.target.size.mb: 10240
supervisor.localizer.cleanup.interval.ms: 600000
```

Empirical observations confirm that systems are becoming increasingly configurable



[Tianyin Xu, et al., “**Too Many Knobs**…”, FSE’15]

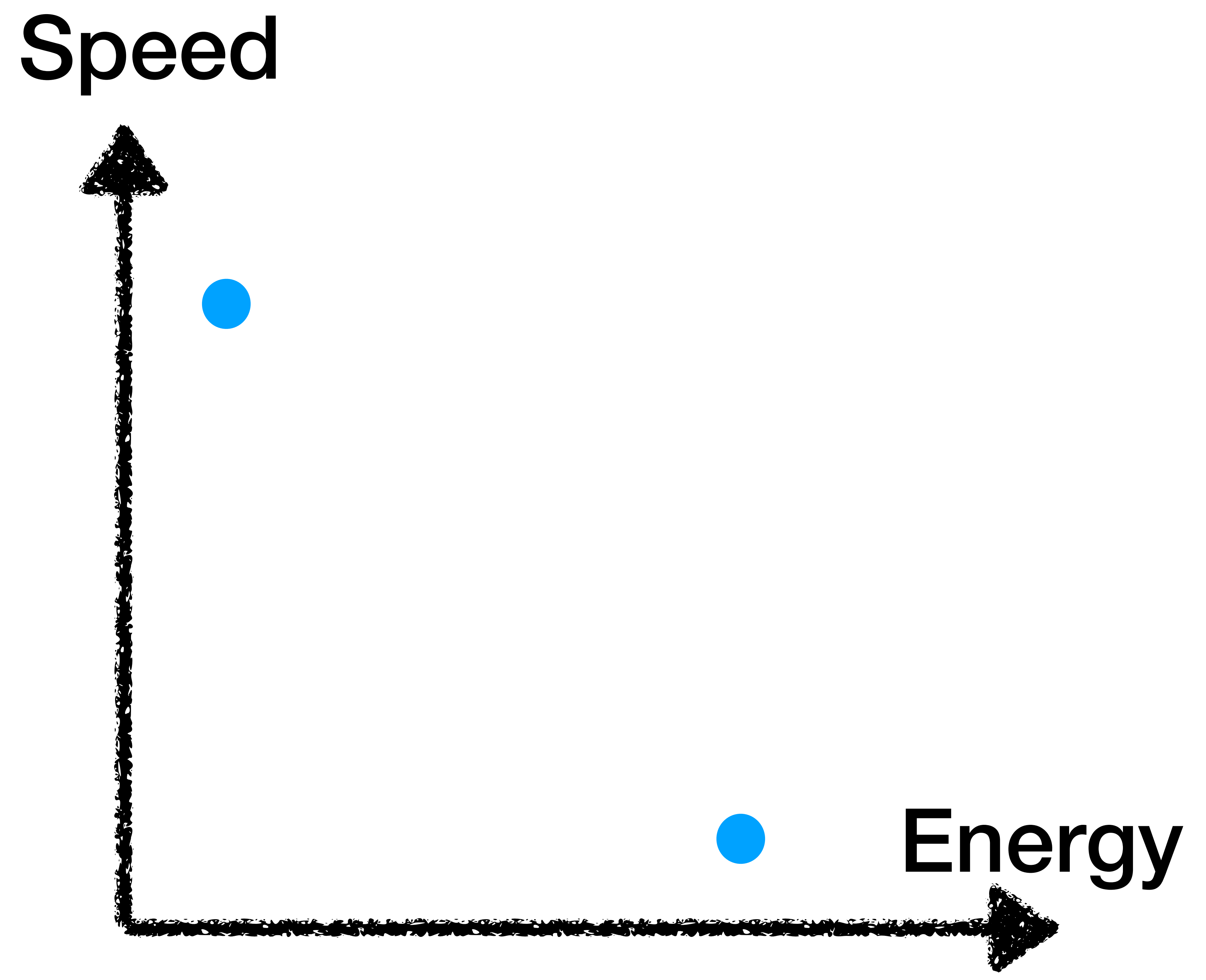
Empirical observations confirm that systems are becoming increasingly configurable



[Tianyin Xu, et al., “**Too Many Knobs**…”, FSE’15]

Configurations determine the performance behavior

```
void Parrot_setenv(. . . name,. . . value){
#ifdef PARROT_HAS_SETENV
  my_setenv(name, value, 1);
#else
  int name_len=strlen(name);
  int val_len=strlen(value);
  char* envs=glob_env;
  if(envs==NULL){
    return;
  }
  strcpy(envs,name);
  strcpy(envs+name_len,"=");
  strcpy(envs+name_len + 1,value);
  putenv(envs);
#endif
}
```



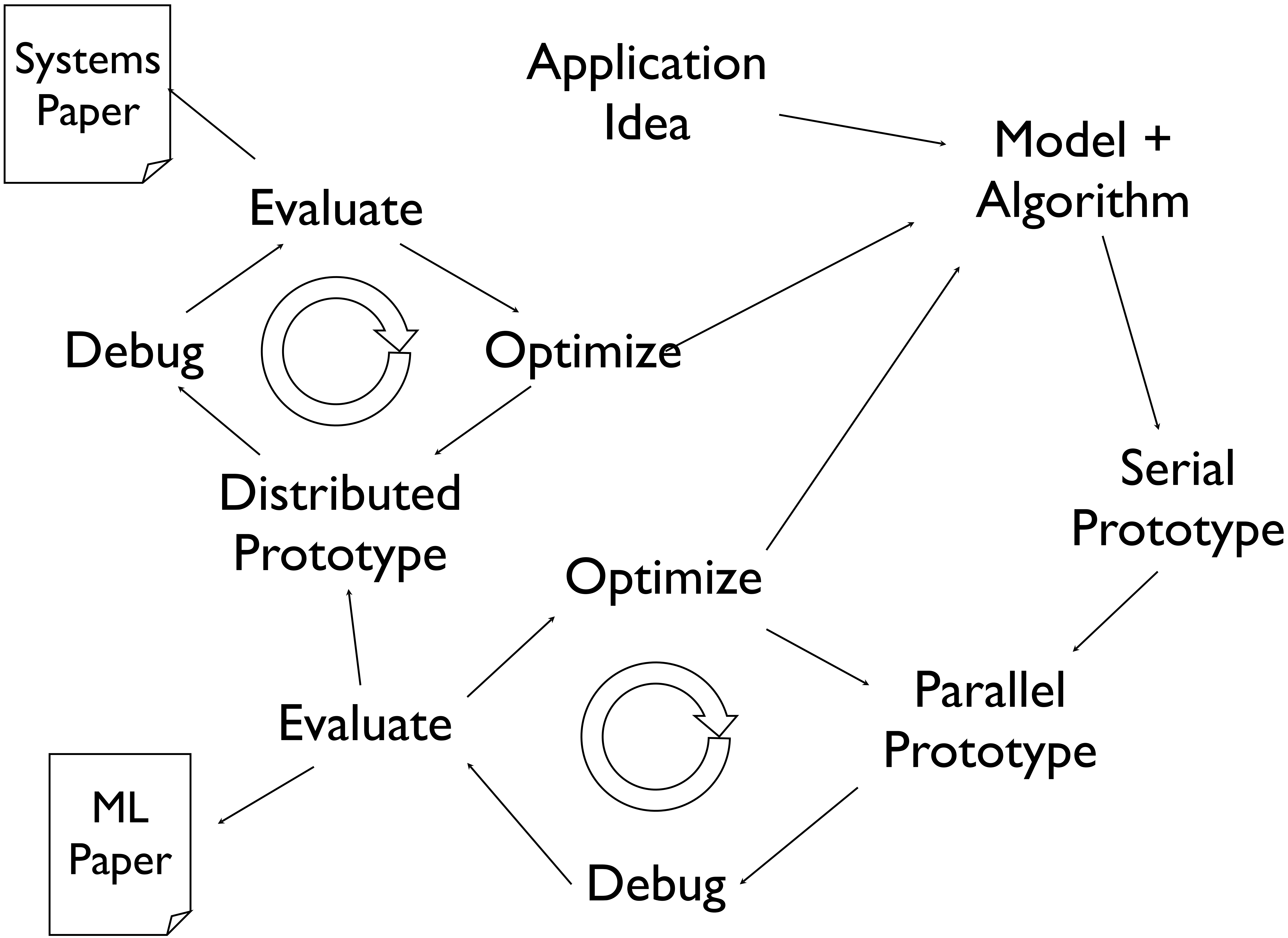
Challenges of configurations

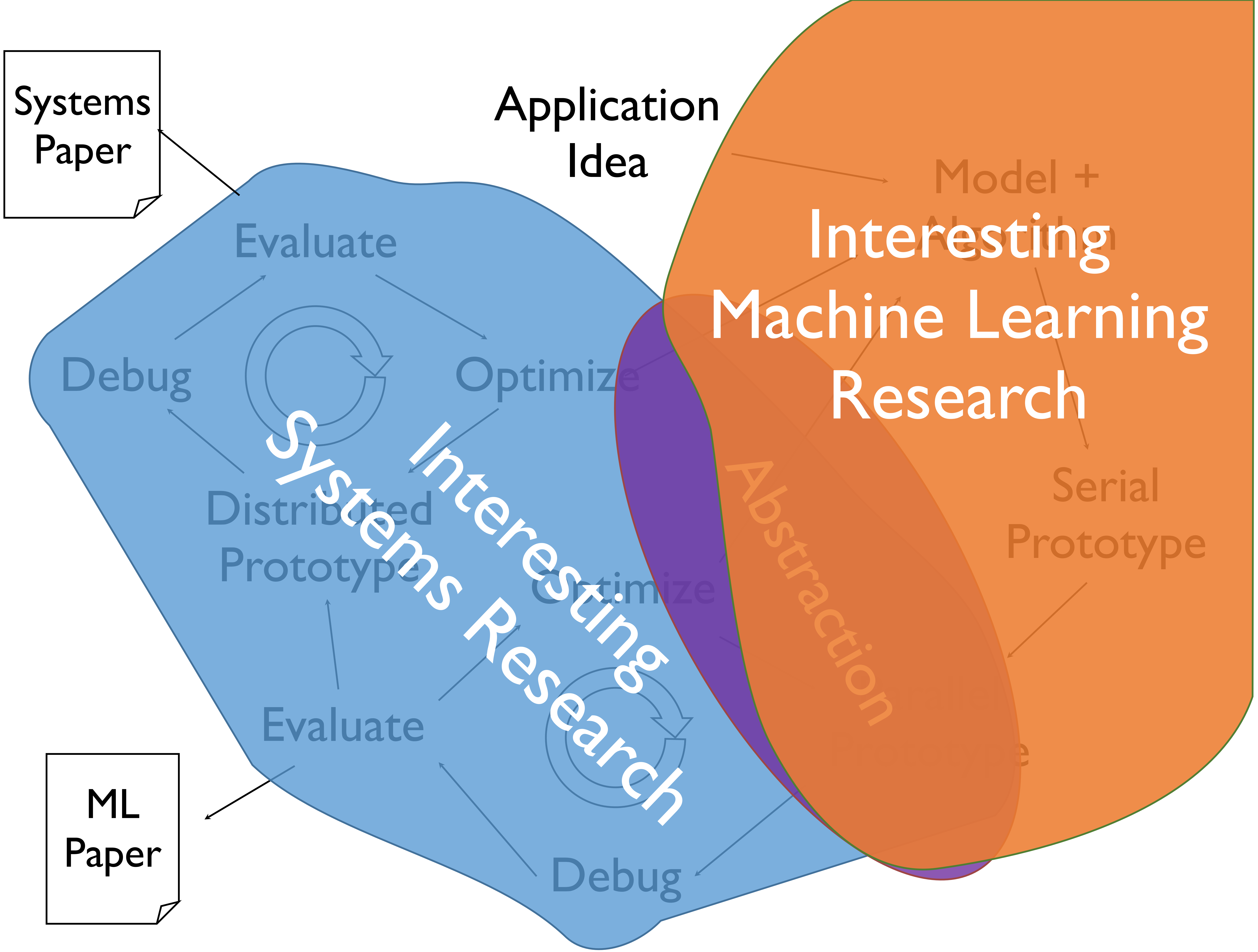
- Difficulties in knowing which parameters should be set
- Setting the parameters to obtain the intended behavior
- Reasoning about multiple objectives (energy, speed)

The goal of my research is…

Understanding the performance behavior of real-world highly-configurable systems that scale well…

… and enabling developers/users to reason about qualities (performance, energy) and to make tradeoffs?


Systems Paper
Application Idea
Model + Algorithm
Evaluate
Debug
Optimize
Distributed Prototype
Serial Prototype
Optimize
Parallel Prototype
Evaluate
ML Paper
Debug


Systems Paper
Application Idea
Model + Algorithm
Interesting Machine Learning Research
Evaluate
Debug
Optimize
Distributed Prototype
Interesting Systems Research
Optimize
Abstraction
Serial Prototype
Evaluate
Debug
ML Paper

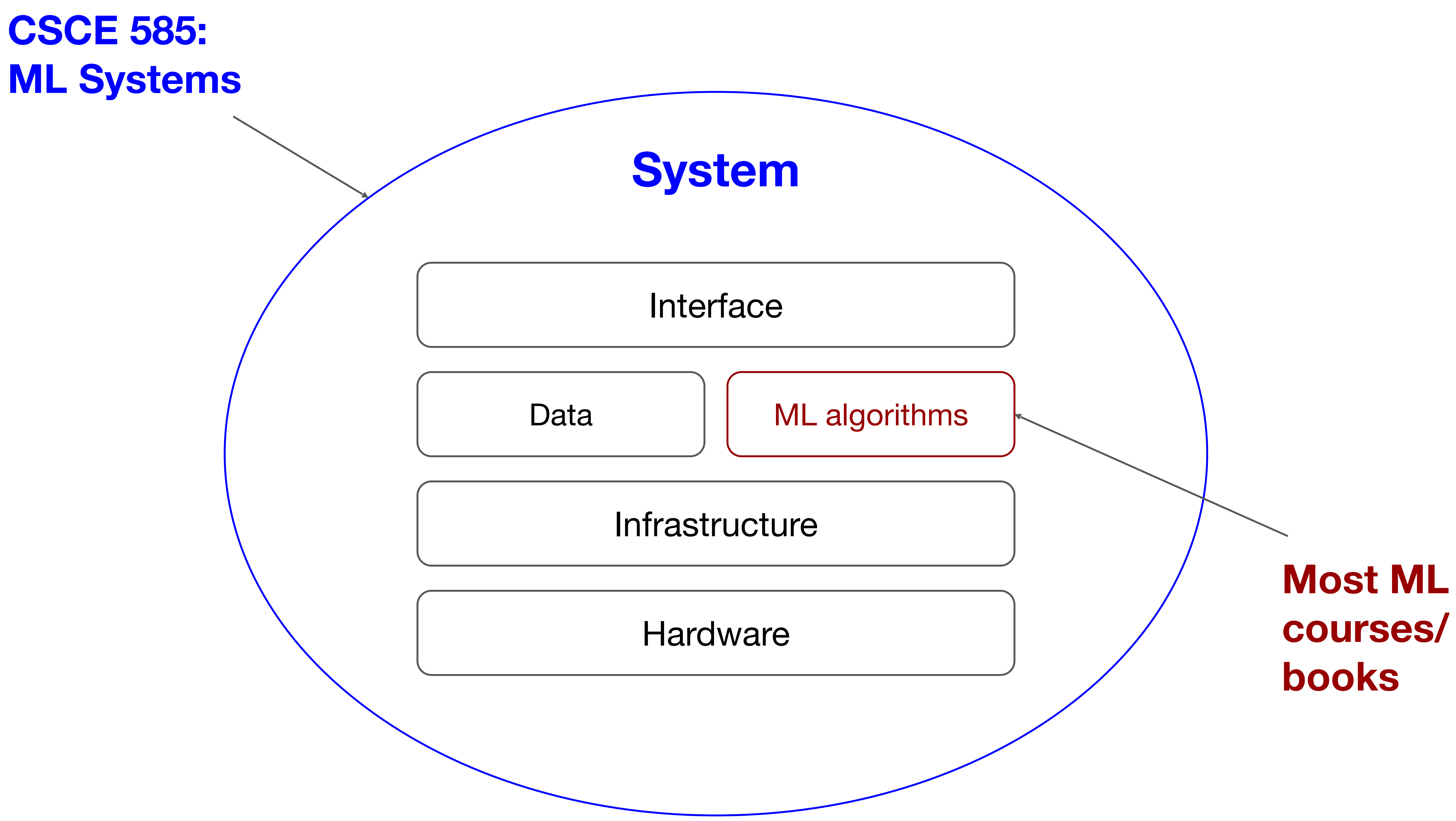
What do you expect to learn in this course?

Why ML Systems instead of ML algorithms?

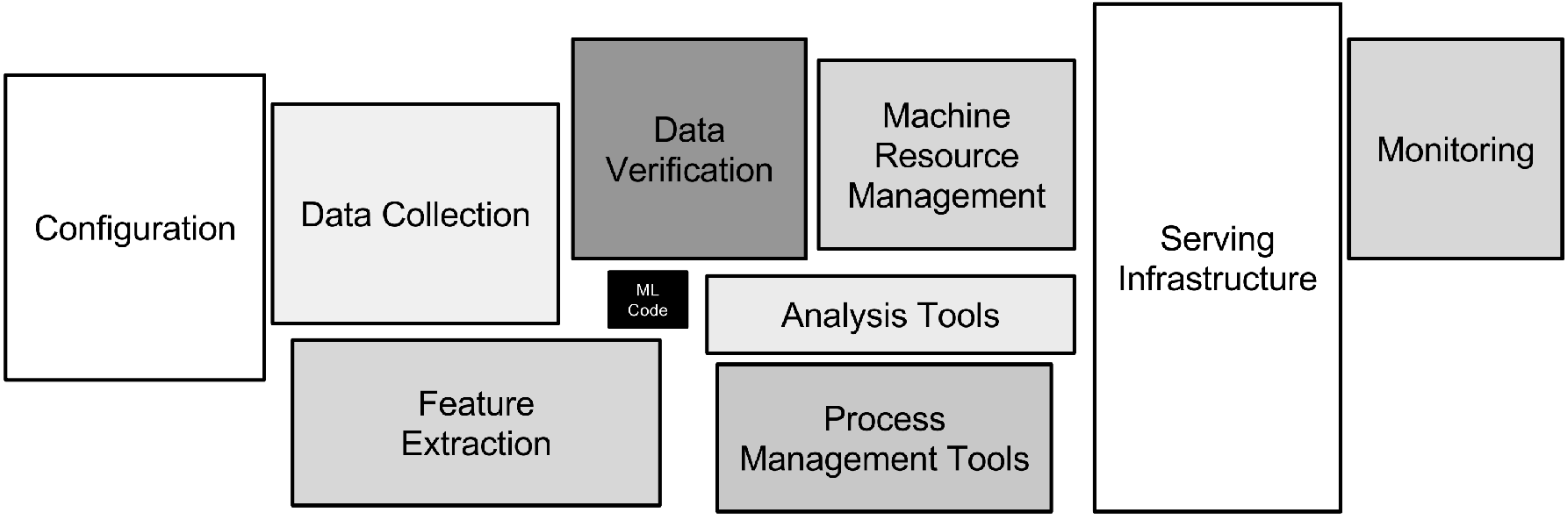
- ML algorithms is the less problematic part.
- The hard part is to **how to make algorithms work with other parts to solve real-world problems.**

Why ML Systems instead of ML algorithms?

- ML algorithms is the less problematic part.
- The hard part is to **how to make algorithms work with other parts to solve real-world problems**.
- 60/96 failures caused by non-ML components


CSCE 585:
ML Systems
System
Interface
Data
ML algorithms
Infrastructure
Hardware
Most ML courses/ books

ML in Production



How are ML systems designed and implemented?

The process of defining the **interface, algorithms, data**, **infrastructure**, and **hardware** for a machine learning system to satisfy **specified requirements**.

What is machine learning systems design?

The process of defining the **interface, algorithms, data, infrastructure**, and **hardware** for a machine learning system to satisfy **specified requirements**.

reliable, scalable, maintainable, adaptable

The questions this class will help answer …

- You’ve trained a model, now what?
- What are different components of an ML system?
- How to do data engineering?
- How to engineer features?
- How to evaluate your models, both offline and online?
- What’s the difference between online prediction and batch prediction?
- How to serve a model on the cloud? On the edge?
- How to continually monitor and deploy changes to ML systems?
- …

This class will cover ...

- ML production in the real world from software, hardware, and business perspectives
- Iterative process for building ML systems at scale
    - project scoping, data management, developing, deploying, monitoring & maintenance, infrastructure & hardware, business analysis
- Challenges and solutions of ML engineering

This class will not teach ...

- Machine learning/deep learning algorithms
  - Machine Learning
  - Deep Learning
  - Convolutional Neural Networks for Visual Recognition
  - Natural Language Processing with Deep Learning
- Computer systems
  - Principles of Computer Systems
  - Operating systems design and implementation
- UX design
  - Introduction to Human-Computer Interaction Design
  - Designing Machine Learning: A Multidisciplinary Approach

Prerequisites

- Knowledge of CS principles and skills
- Understanding of ML algorithms
- Familiar with at least one framework such as TensorFlow, PyTorch, JAX
- Familiarity with basic statistics, linear algebra, and calculus.

You will be fine if you are eager to learn!

Goals for the Class

What do you expect from this class?

Our Goals for the Class

- Identify and **solve impactful problems** at the intersection of AI and Systems
- Learn about the **big ideas** and **key results** in AI systems
- Learn how to read, evaluate, and **peer review papers**
- Learn how to **write great papers** that also get published!

Have Fun!

Reasons not to take this class … ☹

- If you want to learn how to **train models** and **use TensorFlow**, **PyTorch**, and **SkLearn**
- If you want learn how to **use big data systems**
- You are ***<u>not</u>*** interested in learning how to **evaluate**, **conduct**, and **communicate** research
  - Why not?
- You are **unable to attend lecture**

Logistics

Piazza

- Piazza: you have already been added!
- Ask questions
- Answer others’ questions
- Learn from others’ questions and answers
- Find teammates

Weekly Topic Organization

- Each week, we'll cover two topics
- There will be **1 required paper** to read per class, plus optional ones for background
  - **Do the homework** of reading & critiquing the required paper before class!
- **Format of Each Lecture:**
  - **~45 Minute** presentation from **1-2 students** assigned to topic
    - Should cover the required paper and key ideas in optional ones
    - We'll give the presenter(s) instructions for each lecture
    - Signup today! -- Link coming soon.
  - **~45 Minute** class discussion on the strengths, weaknesses, impact, and follow-up opportunities for that area

ML Systems course is project-based

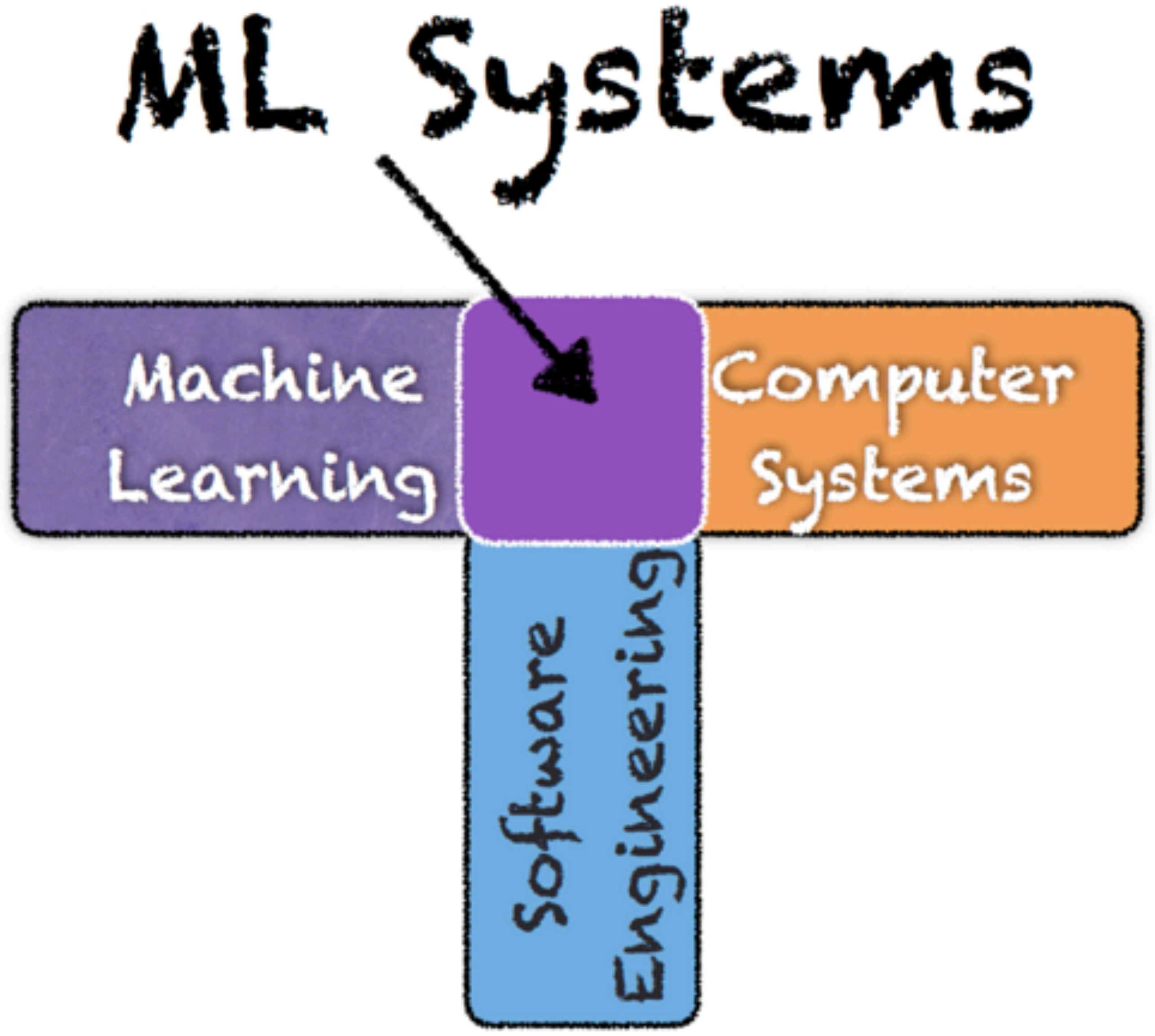
- You can work in teams of up to 2 or 3 people.
- Every team member should be able to demonstrate her/his contribution(s).
- The outcome will be evaluated based on the quality of the deliverables (code, results, report) and presentations/demonstrations.
- The final report contains motivation, positioning in existing literature, technical details, details about the experimental setup, results regarding ablation analyses, comparisons, and conclusions.
- It is essential to write why you designed the experiments in any specific way and how such a design of the experiment would answer any question of hypothesis.

Honor code: permissive but strict - don’t test us ;)

- OK to search about the systems we’re studying.
- Cite all the resources you reference.
    - E.g., if you read it in a paper, cite it.
- NOT OK to ask someone to do assignments/projects for you.
- OK to discuss questions with classmates.
- NOT OK to copy solutions from classmates.
- OK to use existing solutions as part of your projects/assignments. Clarify your contributions.
- NOT OK to pretend that someone’s solution is yours.
- OK to publish your final project after the course is over (we encourage that!)

CSCE 585 LECTURES PROJECTS RESOURCES POLICIES PIAZZA

Machine Learning Systems



New to machine learning? Not sure how ML works in production? Interested to get involved in advanced ML+Systems research? This class is designed for you!

When we talk about Artificial Intelligence (AI) or Machine Learning (ML), we typically refer to a technique, a model, or an algorithm that gives the computer systems the ability to learn and to reason with data. However, there is a lot more to ML than just implementing an algorithm or a technique. In this course, we will learn the fundamental differences between AI/ML as a model versus AI/ML as a system in production.

https://pooyanjamshidi.github.io/mls/

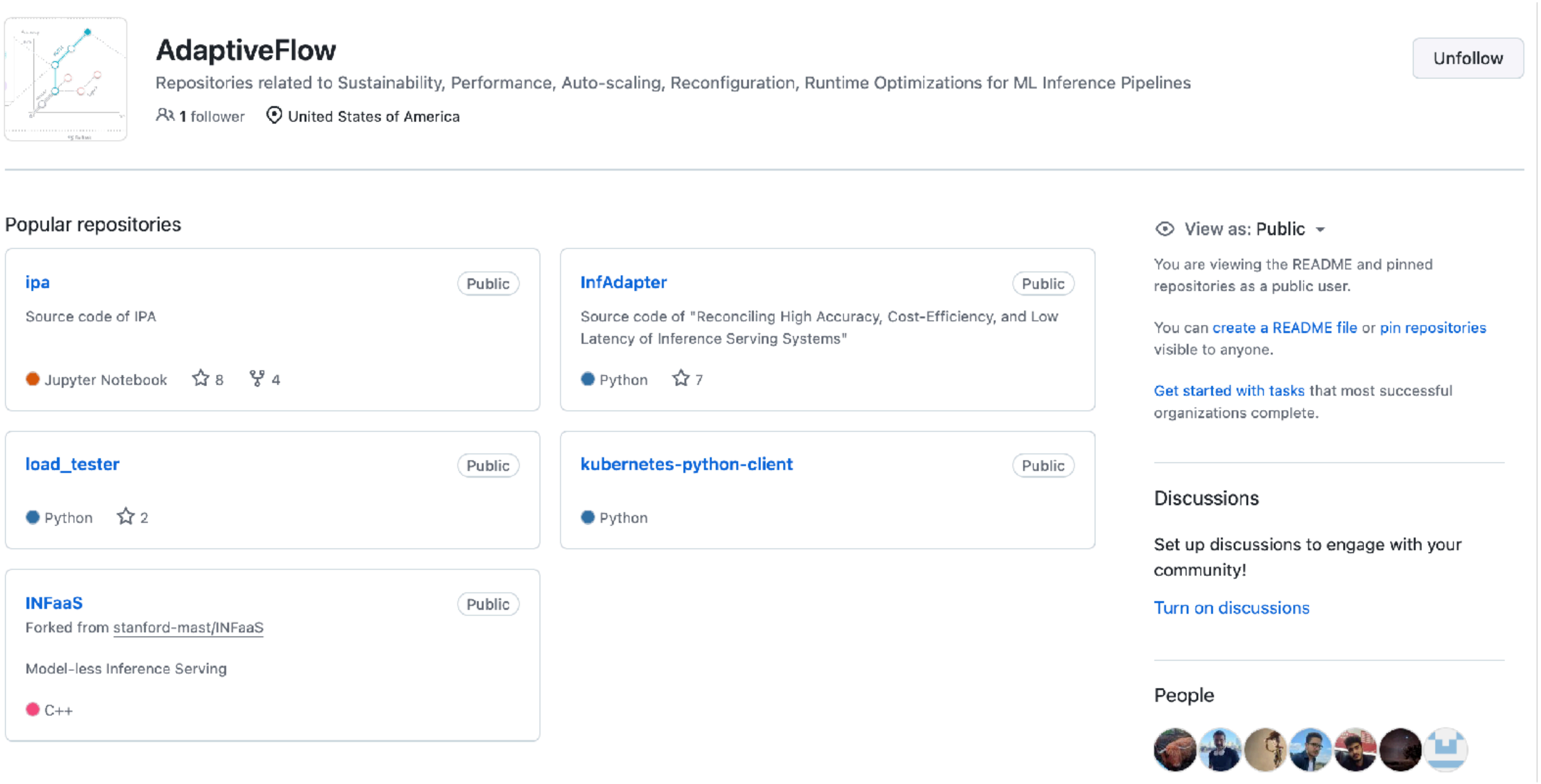
Course Projects

Course Projects

- **Primary Project: ML Pipeline Adaptation:** Configuration Tuning, Runtime Resource Adaptation; Extensions over IPA (https://github.com/reconfigurable-ml-pipeline/ipa)
- **Secondary Options: You define the scope!** Any Relevant Topic to ML Systems; Examples:
  - **LLM Serving**: Batching, Scheduling, Eviction Policy, Prefetching
  - **Robotics:** Multi-Modal Learning and Navigation
  - An **extension** of your current research!
  - A **replication** or an **extension** of existing work.

Primary Project: Extensions on top of the IPA Infrastructure

https://github.com/reconfigurable-ml-pipeline/ipa

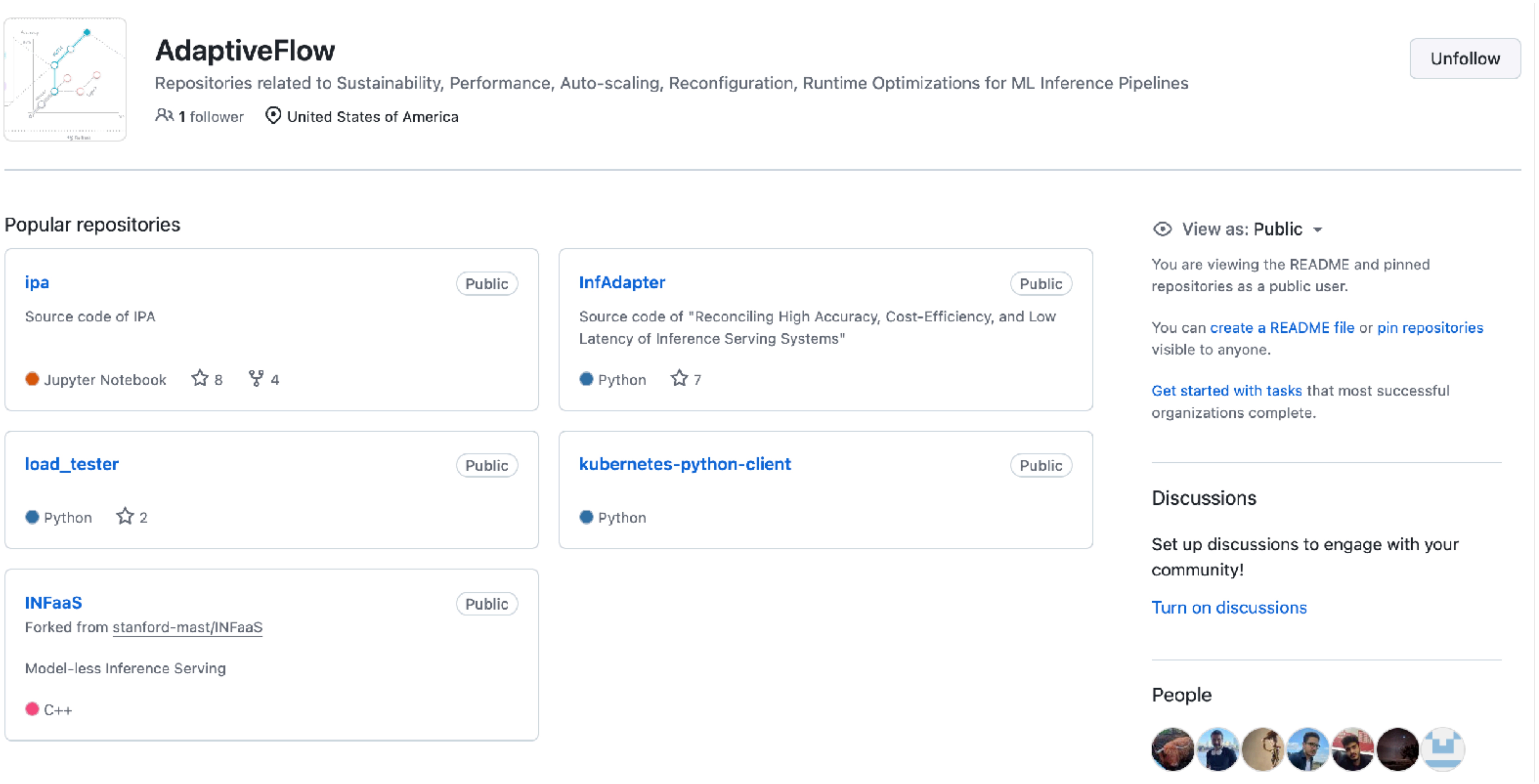


Tools you need for running IPA: https://github.com/reconfigurable-ml-pipeline

Primary Project: Extensions on top of the IPA Infrastructure

Inference Pipeline Adaptation (IPA)

https://github.com/reconfigurable-ml-pipeline/ipa



Tools you need for running IPA: https://github.com/reconfigurable-ml-pipeline

Examples, Tips, Suggestions

Checkout

CSCE 585 LECTURES PROJECTS RESOURCES POLICIES PIAZZA

Projects

Submission

For submitting your homework and project deliverables, please use the instructions and template in the course resources repository.

Topics

The course project is an opportunity to apply what you have learned in class to a problem of your interest. Potential projects must have these two components:

- `Machine Learning` algorithm: Any ML model class including neural networks or any good old-fashioned ML/AI.
- `Computer Systems`: The project should have at least one computer systems component: (i) Platform: Embedded, Realtime, Cloud, IoT, Edge; (ii) Systems issues such as scalability, performance, reliability; (iii) On-device ML: e.g., TinyML, AI on Edge; (iv) Trustworthy AI: Bias, Fairness, Robustness, Privacy, Security, Explainability, Interpretability, Interoperability; (v) Robot Learning, any project that makes robots more intelligent!

Machine learning systems, explained!

The following categories also fit within the scope, and I highly encourage students to consider such projects:

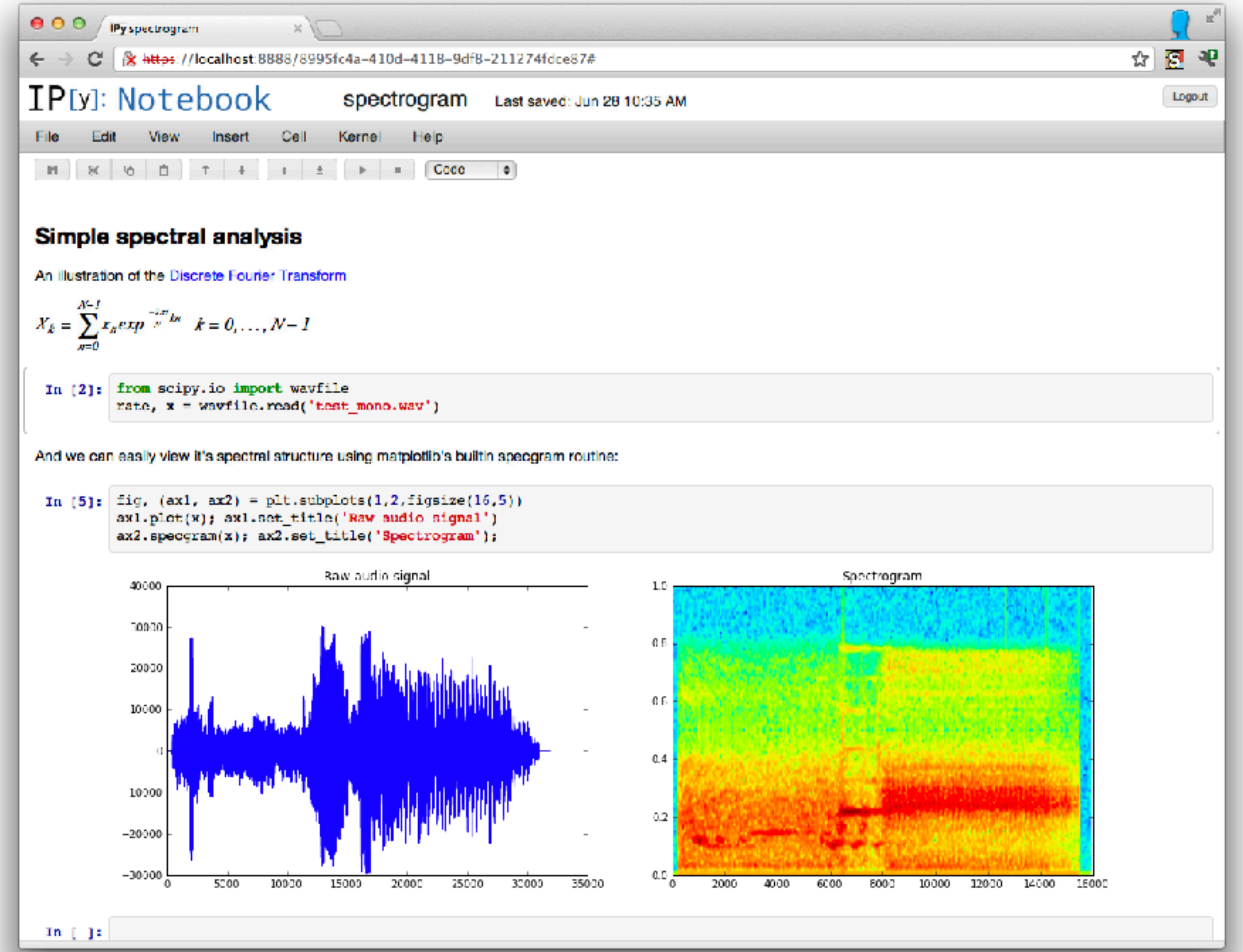
- There are lots of resources that you can read, review, and study to find a specific project idea with a clear scope. For example, you can use `blog posts` from engineering teams at high-tech companies: Uber Engineering, The Netflix Tech Blog, Spotify Labs, Meta, and many more.
- Topics related to `AI in Robotics` or `Autonomy in Robotics` are great for this course. You can use simulators such as Gazebo or Bullet or use cloud services such as Amazon RoboMaker for your project and do not need to have it on a physical robot, but if you want to do a project with physical robots, you can do it in our robotics lab, come and talk with me. I have several project ideas related to Autonomy and Robotics. We have also a robotics team, called Gamecock Robotics. You can define your project to make the robot autonomous. You can also read blog posts to formulate a well-scoped project.
- Building on top of an `open source ML system` that you can find on GitHub, e.g., you can develop a tracking algorithm and develop a plugin for DeepStream

Checkout

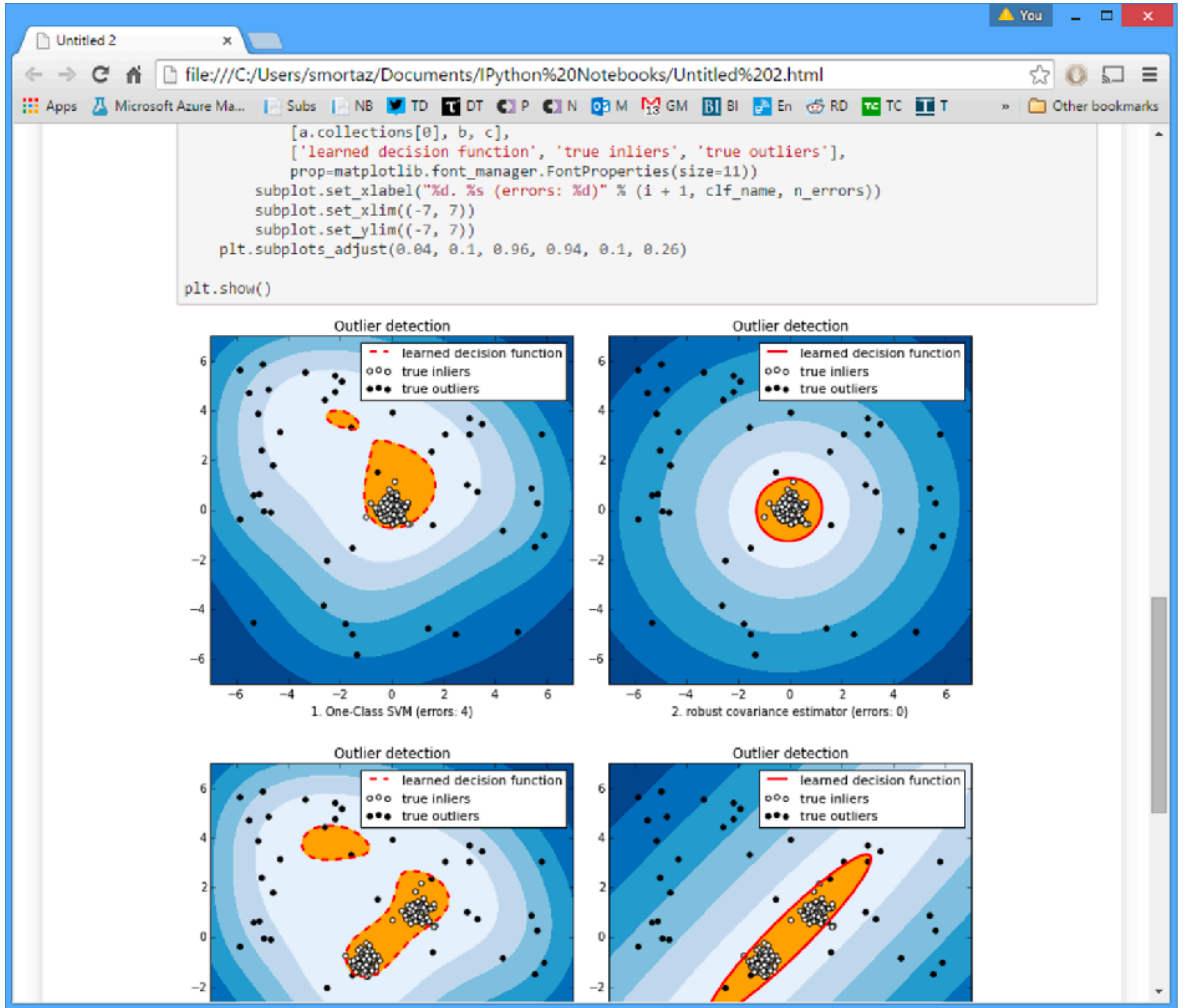
- `AI competitions` are great project ideas for non-CS students, e.g., EvalAI hosts many interesting competitions with prizes suitable for students from all backgrounds, e.g., (i) Open Catalyst Challenge for Chemical Engineering students; (ii) Neural Latents Benchmark for Neuroscience students; (iii) The Robotic Vision Challenges for students interested in robotics.
- `Hackathons` : e.g., PyTorch Annual Hackathon 2021, AWS BugBust, Kaggle
- `Systematic study of open source ML Systems` via (i) interview study (please make sure you design the interview study correctly before conducting the interviews) and/or (ii) Formulating interesting research questions about building ML systems, for example, contrasting testing practices for ML systems vs. traditional software systems, collecting data from software repositories, and systematically extracting info from these repositories that answers your research questions.
- `TinyML` projects: You can find many ideas on GitHub and TinyML community forum
- `AI for Social Good` : There are massive opportunities to define your project on AI for Social Good, just google it and listen to podcasts for ideas, e.g., In Machines We Trust or The TWIML AI Podcast.
- Topics related to `Systems for ML` or `ML for Systems` : There are many opportunities to develop an infrastructure to make the ML workflow faster, more efficient, reliable, dependable, etc. Please check out some of the work at AISys lab.
- `AI and Music` : Topics related to music synthesis, music perception, music ranking, music experience, and many more. Please check out some cool works: (i) OpenAI Jukebox, (ii) Deep Learning Could Bring the Concert Experience Home.
- And, in general, for any interesting ML systems project ideas, try GitHub, or Reddit.

If you are unsure whether the project you have defined fits within the scope, please talk with me after class. If you believe that might be helpful for other students, please ask your question on Piazza or during class hours.

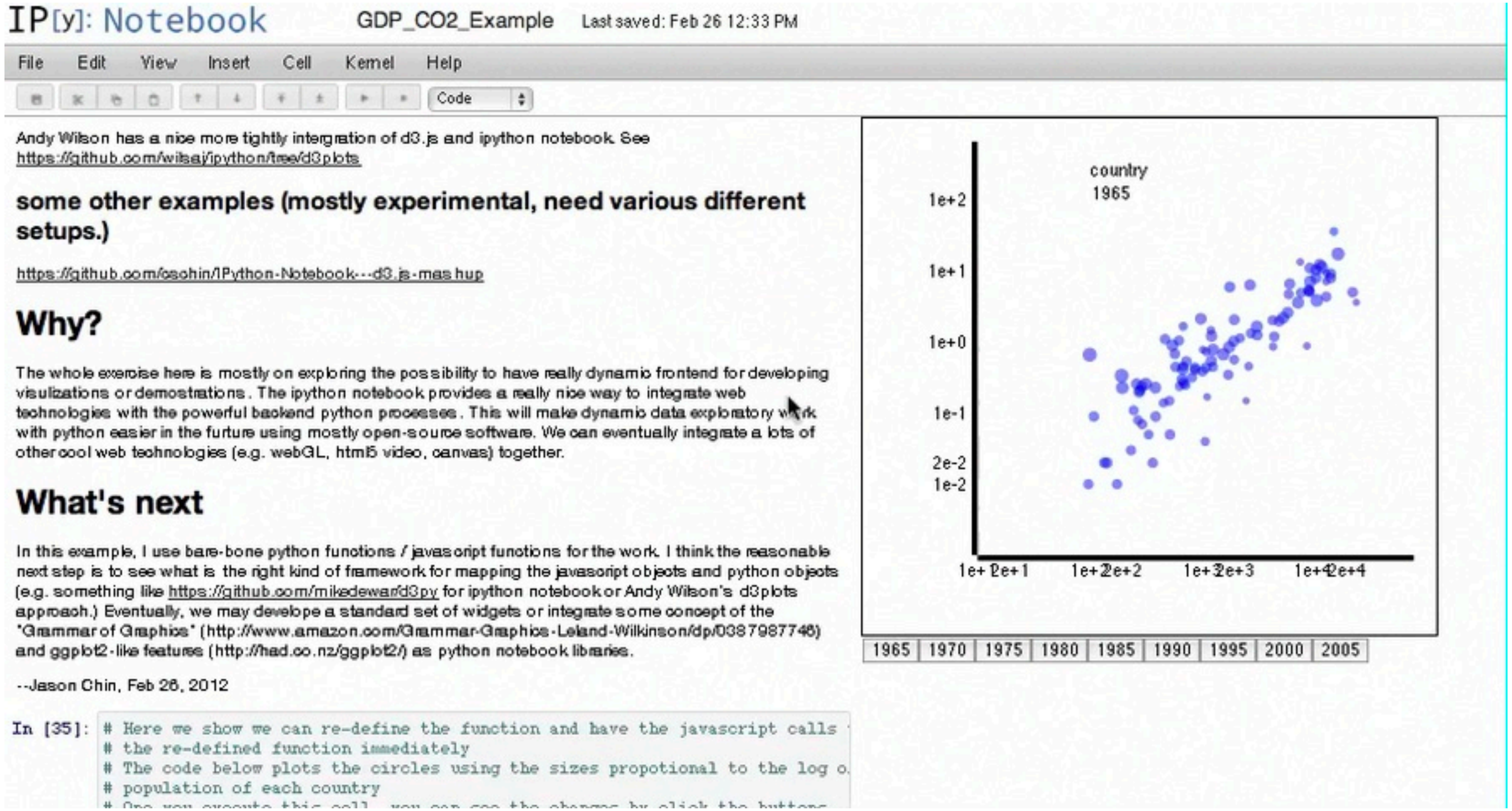
How the project report should looks like?



How the project report should looks like?

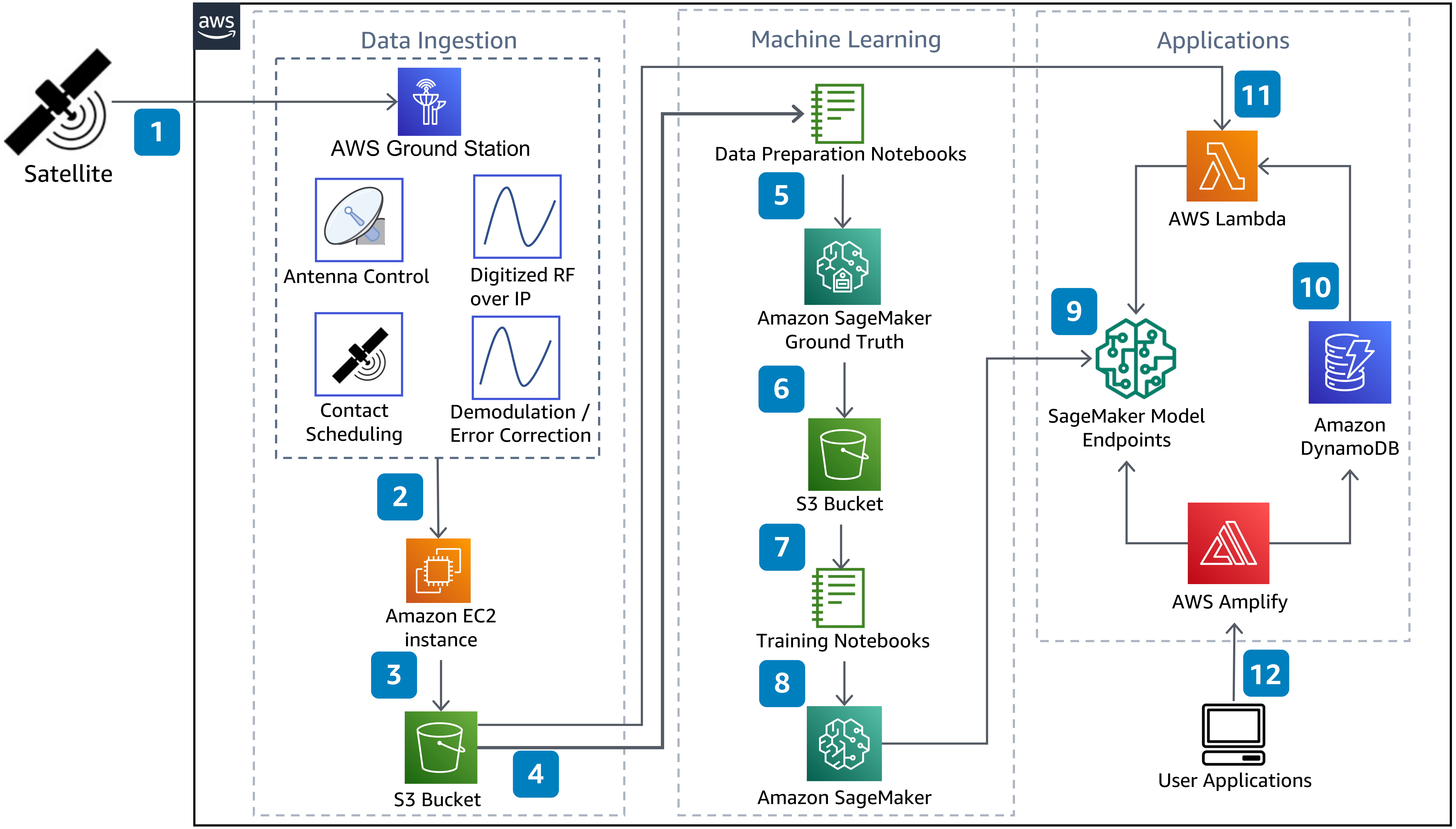


How the project report should looks like?

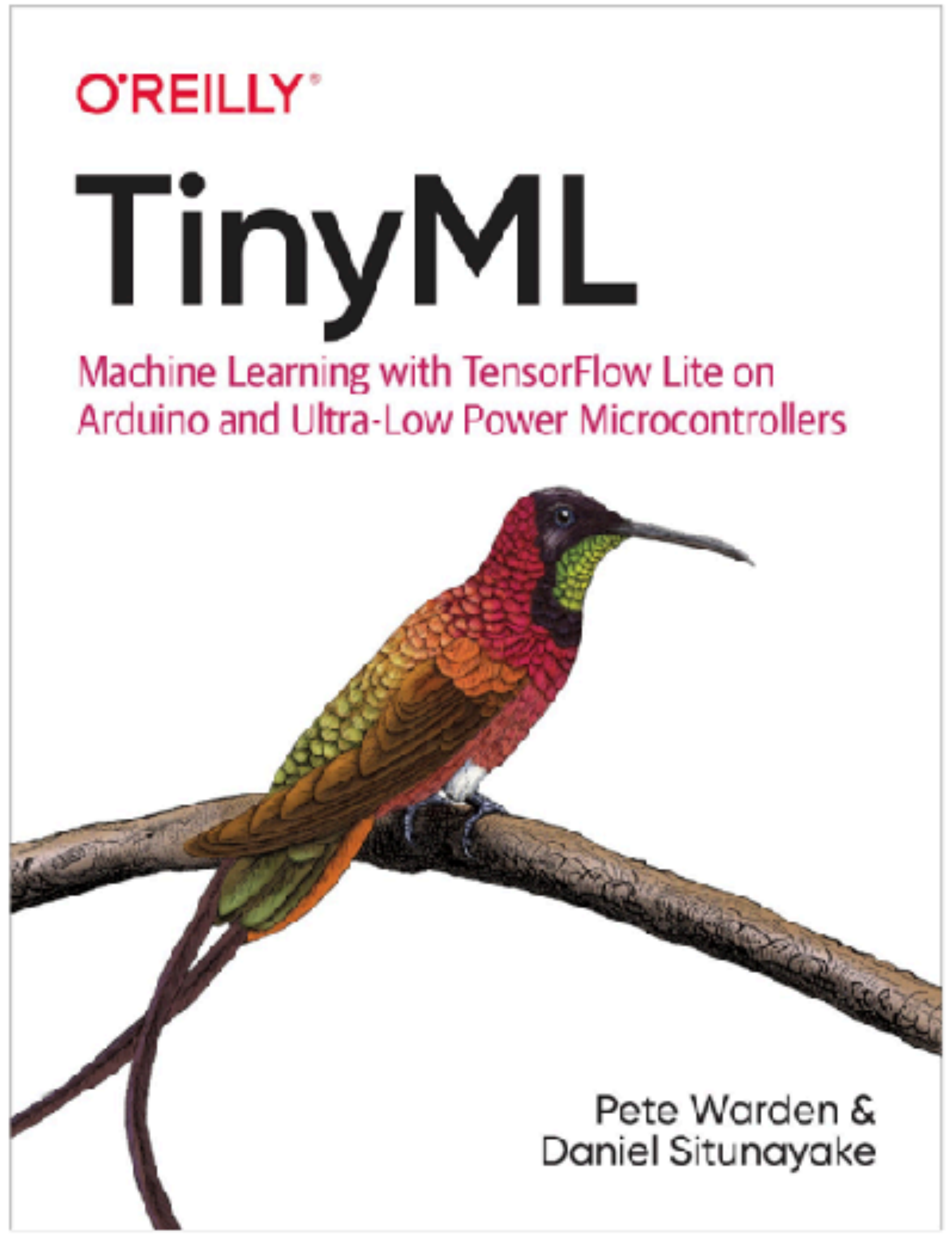


Run Machine Learning Algorithms with Satellite Data

Use AWS Ground Station to ingest satellite imagery, and use Amazon SageMaker to label image data, train a machine learning model, and deploy inferences to customer applications.



1. Satellite sends data and imagery to the **AWS Ground Station** antenna.
2. **AWS Ground Station** delivers baseband or digitized RF-over-IP data to an **Amazon EC2** instance.
3. The **Amazon EC2** instance receives and processes the data, and then stores the data in an **Amazon S3** bucket.
4. A Jupyter Notebook ingests data from the **Amazon S3** bucket to prepare the data for training.
5. **Amazon SageMaker Ground Truth** labels the images.
6. The labeled images are stored in the **Amazon S3** bucket.
7. The Jupyter Notebook hosts the training algorithm and code.
8. **Amazon SageMaker** runs the training algorithm on the data and trains the machine learning (ML) model.
9. **Amazon SageMaker** deploys the ML models to an endpoint.
10. The SageMaker ML model processes image data and stores the generated inferences and metadata in **Amazon DynamoDB**.
11. Image data received into **Amazon S3** automatically triggers an **AWS Lambda** function to run machine learning services on the image data.



Reference Book

We recommend Pete's TinyML book as a reference for the projects and programming assignments. The book is a good primer for anyone new to embedded devices and machine learning. It serves as a good starting point for understanding the machine learning workflow, starting from data collection to training a model that is good enough for deploying on ultra-low power computing devices.

The course builds on top of some concepts covered within this book. We are also preparing an e-book that is a good primer to fill-in material that is supplementary to this book. Stay tuned!

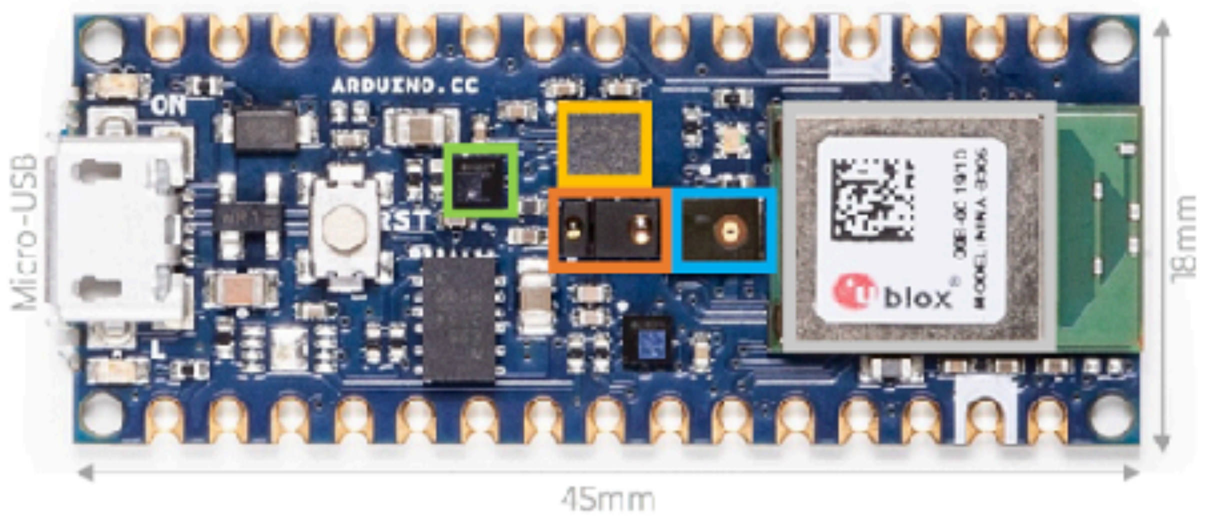
Coding Assignments

To get everyone familiar with coding on embedded systems with ML, we will be using the examples provided in this book as a starting point. Each assignment will build on the examples provided.

Projects

The course will culminate with project demos! You will have an opportunity to showcase what you have learned by incorporating your experience into a hands-on project of your liking. Alternatively, we will provide a list of suggested projects that will allow you to start from the class assignments.

Development Platforms



- Color, brightness, proximity and gesture sensor
- Digital microphone

Cortex-M4 Microcontroller

You will learn to run your ML models on a Nordic nrf52840 processor (256KB RAM, 1 MB Flash, 64 MHz) on the Arduino Nano 33 BLE Sense platform.

TensorFlow Lite

TensorFlow Lite (Micro)

You will use TF Lite (Micro) to deploy your ML models, which is offered free of cost by Google.

Other project ideas

- Focusing on one aspect of ML Systems like testing, deployment, explainability, etc.
- You can work with a company (interview, etc) for documenting their ML practices, then writing a report to be submitted to a conference or a workshop
- Mining software repositories for ML Systems practices (with a central hypothesis)

How to find a good problem for your project?

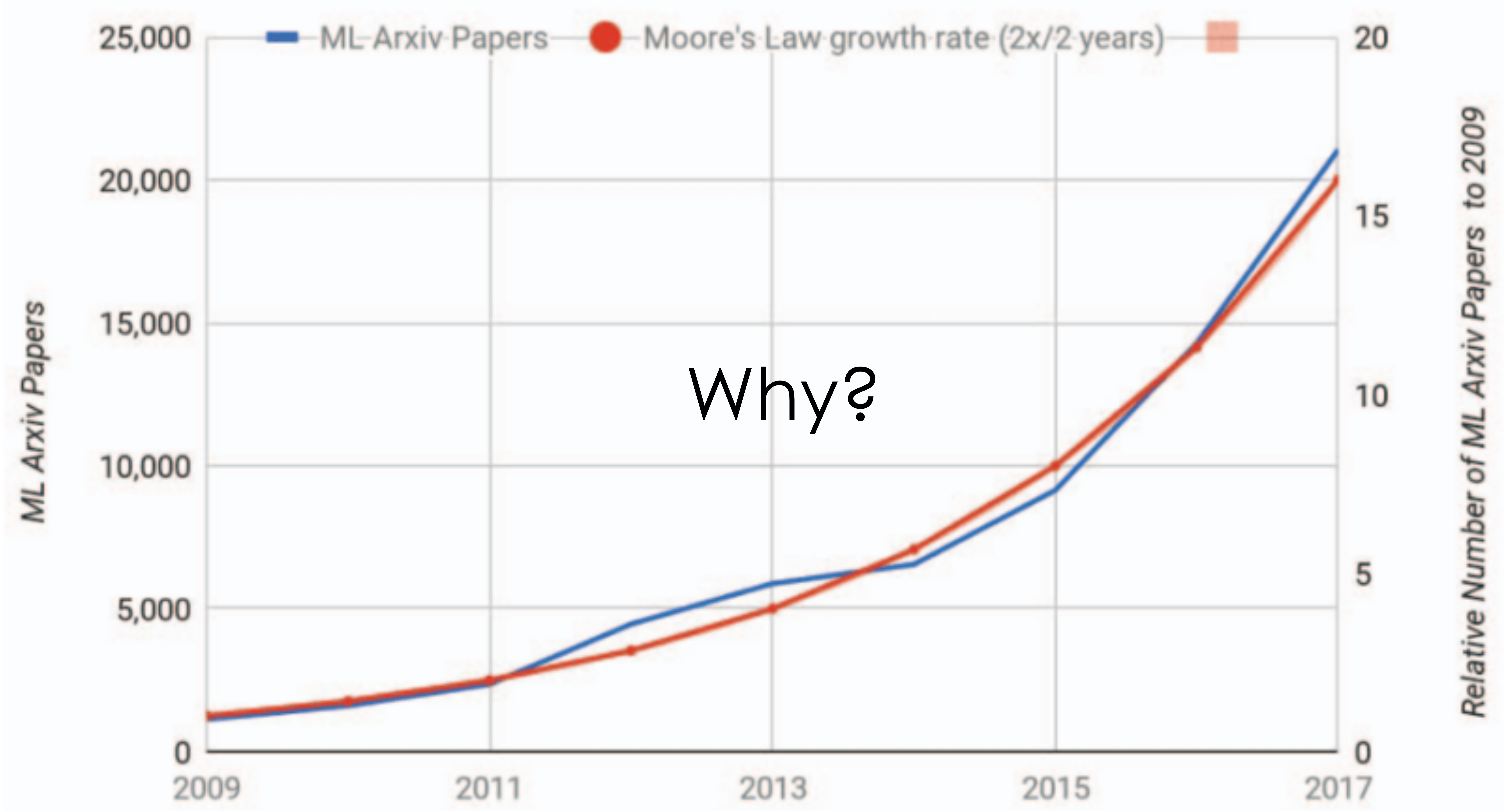
Machine learning community
has had an evolving focus on AI Systems

2023

Large Language Models

This class will focus on **large language models** and the **systems that support them.**

AI Research is Exploding



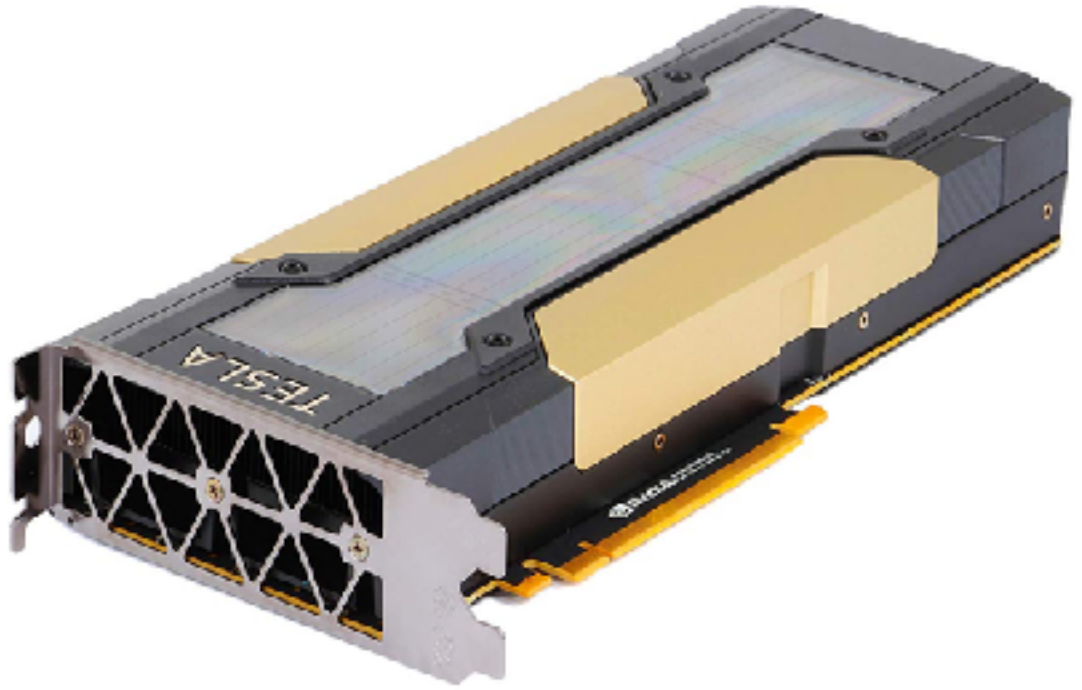
“A New Golden Age in Computer Architecture: Empowering the Machine-Learning Revolution”,
https://ieeexplore.ieee.org/document/8259424

Forces Driving AI Revolution

Data

Benchmarks

Compute

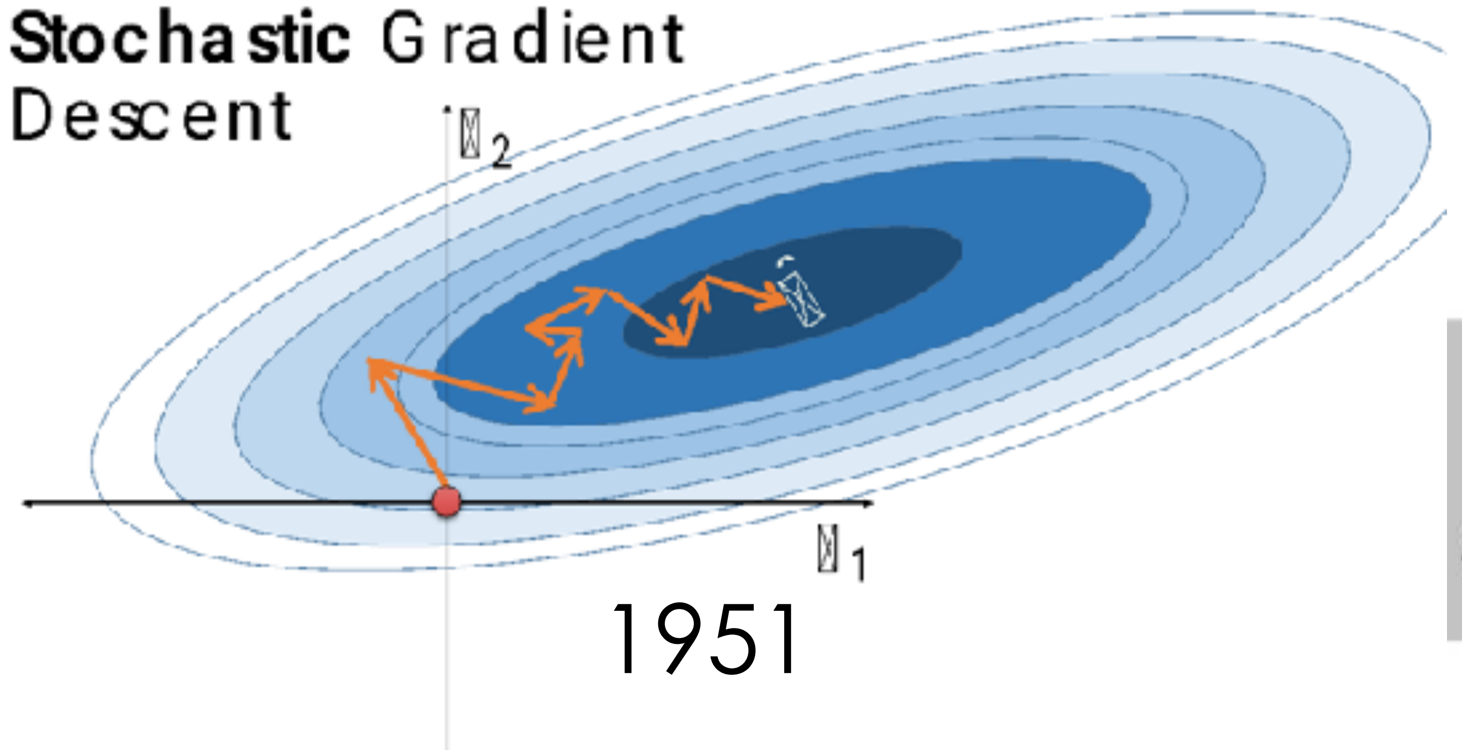


Abstractions

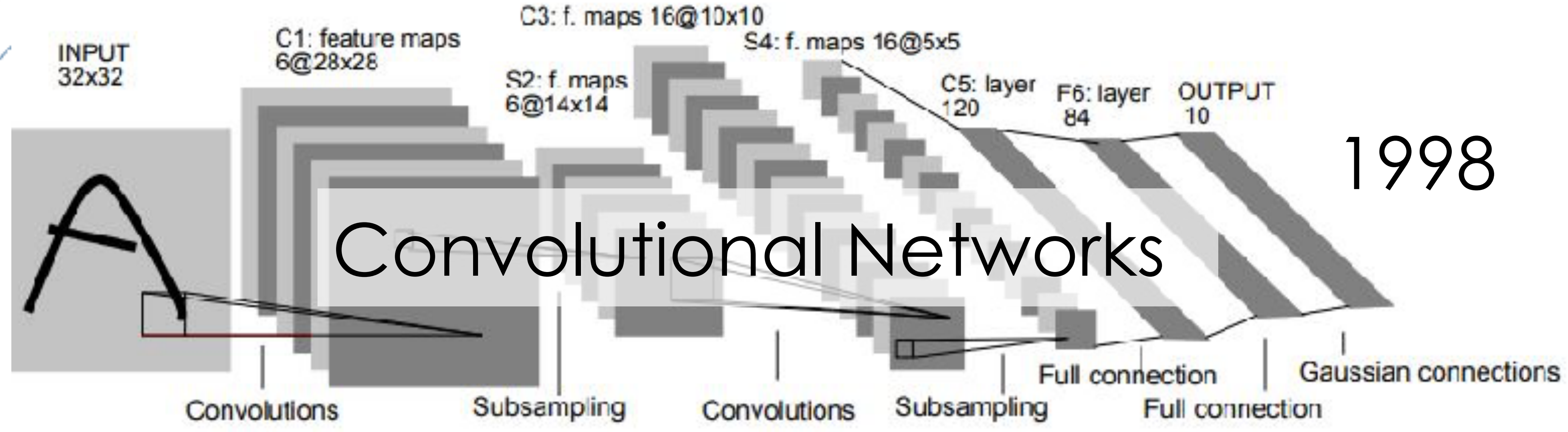
PyTorch

TensorFlow

Advances in Algorithms and Models



1951



Convolutional Networks

1998

What defines good
AI-Systems
Research Today?

What defines good AI-Systems Research Today?

Big Ideas in ML Research

- Generalization (Underfitting/Overfitting)
  - What is being “learned”?
- Inductive Biases and Representations
  - What assumptions about domain enable efficient learning?
- Efficiency (Data and Computation)
  - How much data and time are needed to learn?
- Details: Objectives/Models/Algorithms

What makes a great (accepted) paper?

- **State of the art results**
  - Accuracy, Sample Complexity, Qualitative Results **…**
- **Novel settings,** problem formulations, and **benchmarks**
- Innovation in **techniques**: architecture, training methodology, …
- **Theoretical results** that provide a deeper understanding
- **----**
- **Narrative** and **framing** in prior work and **current trends?**
- **Parsimony?** Are elaborate solutions rejected? If they work better?
- **Verification** of prior results?

What defines good AI-Systems Research Today?

Big Ideas in Systems Research

- **Abstraction**
  - Designing useful building blocks that hide complexity from application developers (modularity, layering, optimization, etc)
- **Tradeoffs**
  - What are the fundamental constraints?
  - How can you reach new points in the trade-off space?
- **Problem Formulation**
  - What are the requirements, assumptions, and goals?

What makes a great (accepted) paper?

- **State of the art results**
  - Simplicity, throughput, latency, reliability, cost, scale, …
- **Generality** of the proposed problem or solution
- **Novelty** of problem formulation, solution, or benchmarks
- Innovation in **techniques**
  - Algorithms, data-structures, policies, software abstractions.
- What you **remove** or **restrictions** also often important
- **Narrative** and **framing** in prior work and **current trends?**
- **Open source?** Real-world use?

What defines good AI-Systems Research Today?

What is AI-Systems Research?

- Good AI and Systems research
  - Provides **insights to both communities**
  - **Builds on big ideas** in prior AI and Systems Research
- Leverages understanding of both domains
  - Studies **statistical and computational tradeoffs**
  - Identify **essential abstractions** to bridge AI and Systems
  - Reframes **systems problems as learning problems**
- More than just useful open-source software!
  - But software impact often matters…

Problems

What makes a good problem?

Joey

What makes a **good problem?**

- **Impact:** People care about the solution
  - … and progress advances our understanding (**research**)
- **Metrics:** You know when you have succeeded
  - Can you **measure progress** on the solution?
- **Tractable:** The problem can be divided into smaller problems
  - You can identify the first sub-problem.
- **Your Edge:** Why is it a good problem ***for you?***
  - Leverage your strengths and imagine a new path.

Can you Solve a Solved Problem?

- Ideally, you want to solve a **new** and **important** problem.
- A **new solution** to a solved problem can be impactful if:
  - It supports a **broader set of applications** (users)
  - It **reveals a fundamental trade-off** or
  - Provides a **deeper understanding** of the problem space
  - **10x Better?**
    - Often publishable…
    - Should satisfy one of the three above conditions … as well

Systems Challenges

What matters for making ML Systems fast, scalable, etc?

Performance

- The elephant in the room given “scale is all you need”!
- Lots of **levels** to work at:
  - Model architecture
  - Algorithms
  - Software optimization (mostly instructions & data movement)
  - Hardware
  - Larger systems (e.g. a datacenter serving multiple models)

What Matters for System Performance?

Table 1. Speedups from performance engineering a program that multiplies two 4096-by-4096 matrices. Each version represents a successive refinement of the original Python code. "Running time" is the running time of the version. "GFLOPS" is the billions of 64-bit floating-point operations per second that the version executes. "Absolute speedup" is time relative to Python, and "relative speedup," which we show with an additional digit of precision, is time relative to the preceding line. "Fraction of peak" is GFLOPS relative to the computer's peak 835 GFLOPS. See Methods for more details.

Version	Implementation	Running time (s)	GFLOPS	Absolute speedup	Relative speedup	Fraction of peak (%)
1	Python	25,552.48	0.005	1	—	0.00
2	Java	2,372.68	0.058	11	10.8	0.01
3	C	542.67	0.253	47	4.4	0.03
4	Parallel loops	69.80	1.969	366	7.8	0.24
5	Parallel divide and conquer	3.80	36.180	6,727	18.4	4.33
6	plus vectorization	1.10	124.914	23,224	3.5	14.96
7	plus AVX intrinsics	0.41	337.812	62,806	2.7	40.45

Leiserson et al, There's Plenty of Room at the Top, Science, 2020

What Matters for System Performance?

- The most expensive thing physically is **data movement**, so that tends to matter most
  - Processor core to cache
  - Cache, to other cache levels, to memory
  - Memory to storage & network
  - This is why we worry about batching, block sizes, precision, etc!
- Of course, other things matter too:
  - **Utilization:** keep all parts of system working all the time
  - **Amdahl's Law:** diminishing returns from optimizing 1 component
  - **Instruction count**

Reliability

- Large, parallel systems are more likely to experience failures and stragglers
- Lots of techniques to reduce **rate** and **impact** of failures
  - Generally easier to optimize mean time to repair (MTTR) than mean time between failures (MTBF)
- ML gives more flexibility because it’s approximate

Programming Models

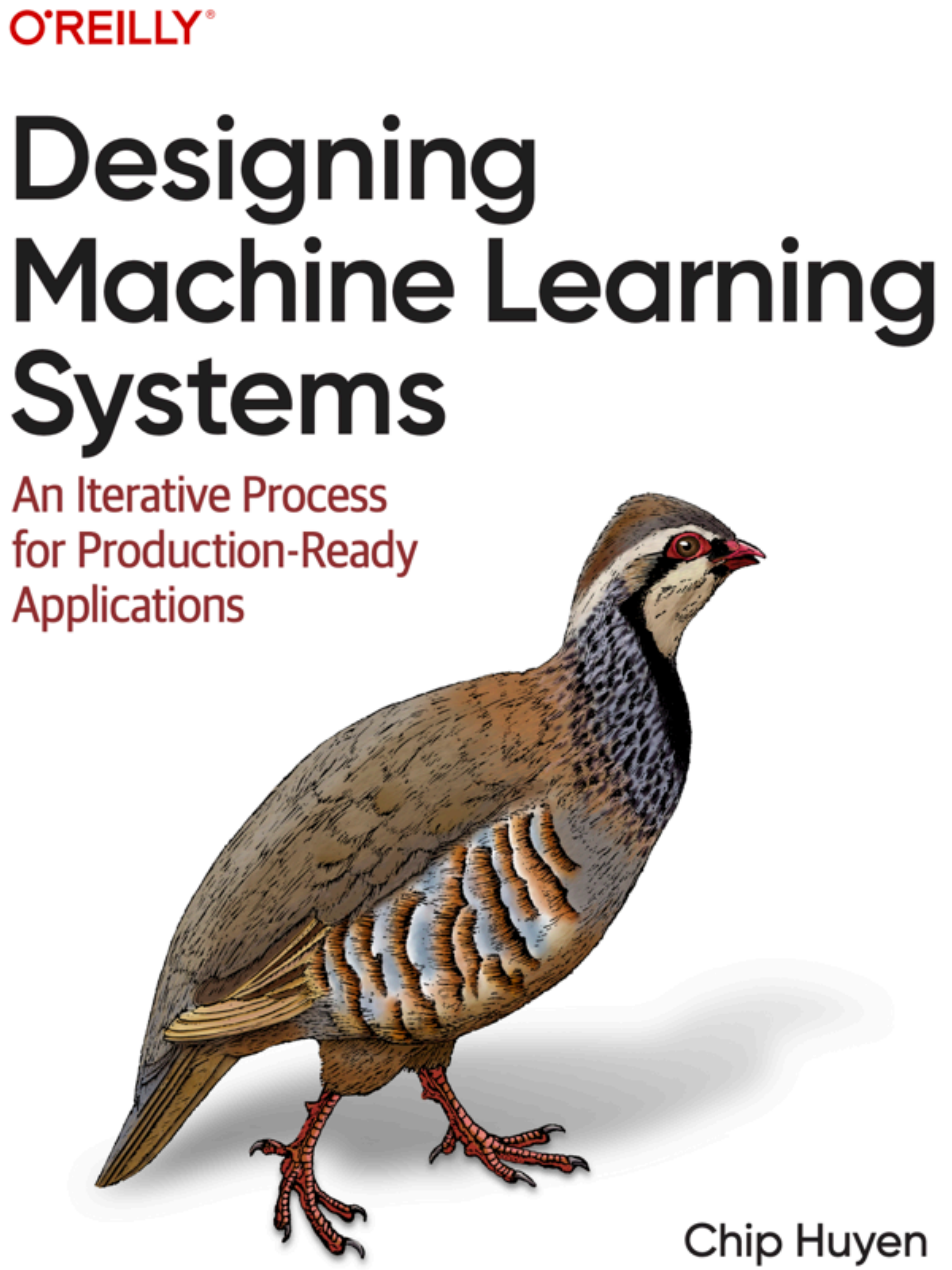
- What are good abstractions to build AI applications?
  - **Upper level:** integrating models into a larger app, tracking and improving app quality over time
  - **Middle level:** building model training and inference software
  - **Lower level:** building math kernels, compilers, etc
- A good abstraction successfully frees the programmer of one or more concerns (e.g., performance or failures) while supporting a wide range of apps on top

Security

- Also a “systems” concern about abstractions and their potential misuse
- With DNNs today, one can fairly easily:
  - Recover the training data that went into models
  - Find adversarial inputs that push models to a target output
  - Bypass current methods to *train* models to be safe
- LLMs are especially troublesome:
  - They sort of do “code execution” (instruction following)
  - People are trying to let them take actions (plugins, agents)

Learning Materials

Course Textbook



Learning Materials

- Learn one of these frameworks: TensorFlow, PyTorch, JAX
  - There are many good tutorials for each framework on their website
- Try to build some very simple models with available benchmarks
  - Search for the LeNet model and train it with the MNIST dataset
  - Try to feed some input data and get the prediction and print it on the console
  - Then try to measure basic performance metrics such as Accuracy or Inference time

Assignments

Assignment #1 (5 minutes presentation)

Due: This Thursday, September 5

- You present your project idea in **3 minutes** (only 3 slides!)
- We discuss and give feedback in **2 minutes.**
- So, make the best out of these 5 minutes to get the most useful feedback for your project proposal.
- We should finish all presentations on Thursday, so please make sure you stick with the time. If you need to leave by the end of the class, please make sure you volunteer to present early, and if we cannot finish all presentations by the end of the class, we will move to another class to finish! If any team members cannot attend, please make sure other members are available to present. No excuse!

Slide #0: Project Title

that represents your project

- Indicate a **title** for your project that represents your project.
- Provide a list of **members** in your team ideally with their headshot, name, their major, and their role in the project.
    - I would like to hear why you teamed up with each other.
    - Also, I would like to hear how you are going to synch with others, in what regularity, etc. I would like to see a good synergy between you right at the beginning.
- Please make sure everything in your slides is high quality (high-resolution pictures, not many words, and clean).

Slide #1: Problem

What is the problem that you will be investigating?

- **Why** is it interesting? **Why** is it interesting to you?
- Please indicate your **project type**:

  1. A replication and (extension) of existing work (including IPA or whatever paper you chose)
  2. An extension of your current research work (in this case, you should indicate where are you in your current research and what will be the new stuff, you should also talk with your advisor)
  3. A real problem that you know is real, and you can talk about it and convince me and others that it is a real problem ideally you know people that faced the problem and would be happy to use your system/tool when you develop it.

Slide #2: Solution

What solution are you proposing?

- Is it a new method, new algorithm, or new tool? What will be the **key outcome**/ deliverable?

- Ideally, this slide should contain a **nice figure to contextualize** your solution.

- If there are existing implementations, will you use them, and how? Ideally, annotate them in the figure. How do you plan to improve or modify such implementations? You don't have to have an exact answer at this point, but you should have a general sense of how you will approach the problem you are working on.

Slide #3: Evaluation

- How will you **evaluate** your results?
  - Qualitatively: what kind of results do you expect (e.g. plots or figures)?
  - Quantitatively: what kind of analysis will you use to evaluate and/or compare your results (e.g., what performance metrics or statistical tests)?
  - What are the existing solutions that you want to compare against them (if you know them already)? This does not have to be comprehensive at this point, but this list will be updated based on how your project will evolve throughout the semester.

Feedback

- Please make sure one member acts as a scribe for the team.
- The scribe would write all feedback.
- You need to write down all verbal feedback in written format throughout all your presentations in the class.
- Because, when you submit, they must be part of your submission by indicating how you addressed the major feedback/comments.

Assignment #2: Project Proposal

Due: Tuesday Sept 10

- This assignment is easy, simply turn your presentation into a written format (.md file in GitHub).
- Please list the **comments** you have received in a section right at the beginning.
- Address the **feedback** in your project proposal.
- Add any **references**, at least you must have a couple of key references.
- If there is a **risk**, indicate it in your proposal, and a rough plan of how you are going to mitigate it.
- Please make sure you list your **project repository** in the Excel sheet that I shared on Piazza.

What if we cannot submit an assignment by the due date?

- Well, do not worry as long as you have tried and have a concrete plan to finish it.
- Simply **ask for an extension over Piazza** and indicate what you have done and what is your plan to finish it up.
- **Example**: We have finished this and that sections, but we need to read this one more paper to get a better idea about the evaluation. Could we submit it on this date?
    - My default answer to such requests would be: you have got it!
    - So, even if I did not get back to you, assume you have got it as long as you provided a good reason for it, but please please do not ask for an extension, if you do not have a good reason for it.
- I always care about learning, please do not worry about your grades, but do worry about learning and try your best to **maximize your learning!**