CSCE 585: Machine Learning Systems

Pooyan Jamshidi

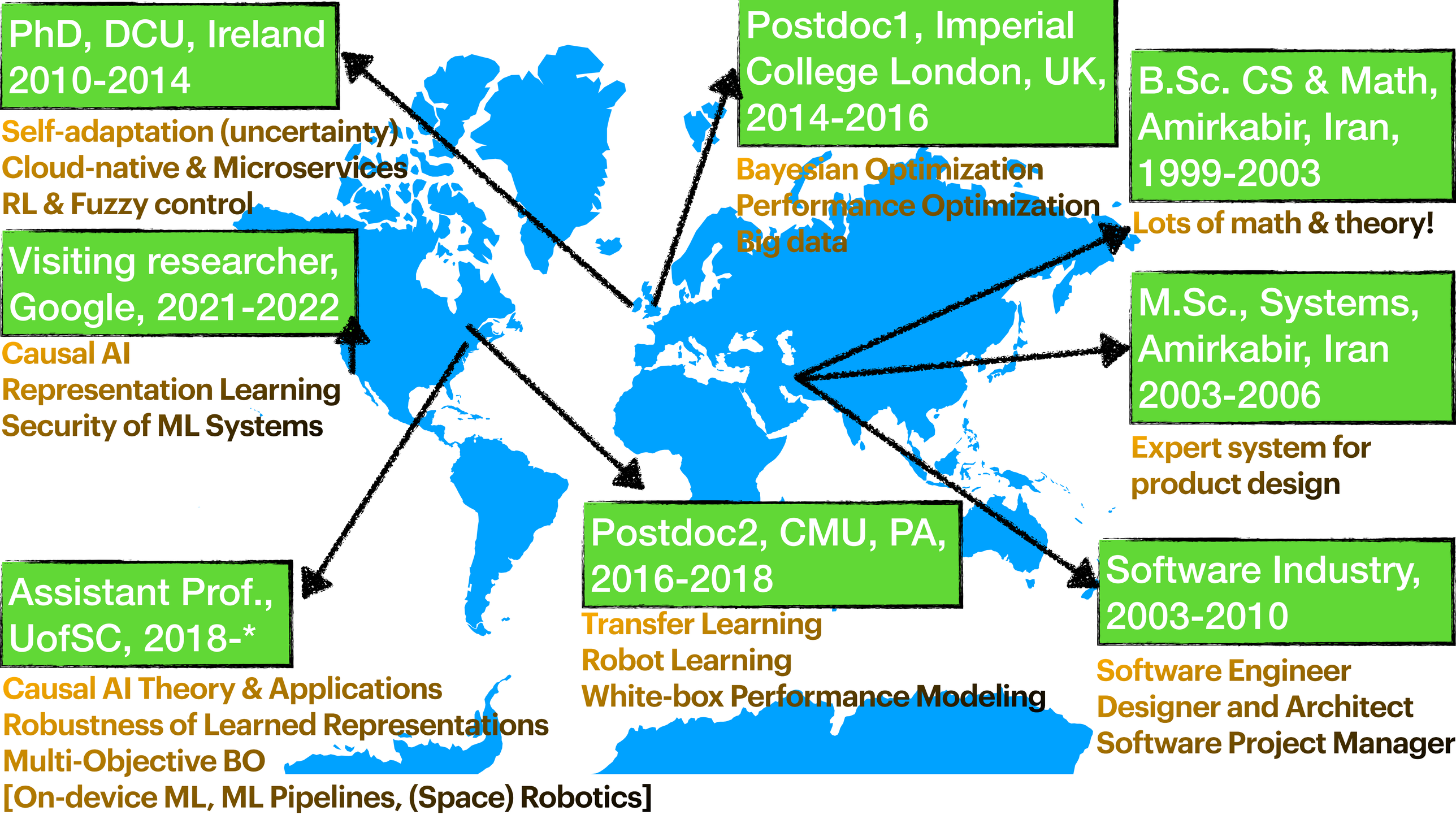
Fall 2024 | Course Website: https://pooyanjamshidi.github.io/mls/

UNIVERSITY OF South Carolina

A brief self introduction

- Assistant Professor @ USC (CEC, CSE), since August 2018
- Postdoc 2 @ Carnegie Mellon University (US), 2016 - 2018
- Postdoc 1 @ Imperial College London (UK), 2014 - 2016
- Ph.D. from Dublin City University (Ireland), 2010 - 2014
- M.Sc. from Amirkabir University of Technology (Iran), 2006
- B.Sc. from Amirkabir University of Technology (Iran), 2003
- Worked at Google and NASA
- https://pooyanjamshidi.github.io/ pjamshid@cse.sc.edu

- Research and Teaching in:
  - Machine Learning Systems = AI/ML + Computer Systems
  - Autonomous Robots = AI/ML + Robotics
  - Causal AI = Causal Inference, Causal Representation Learning
  - Neural Architectures + Hardware Accelerators
  - Systems for ML (See CSCE 585)
  - Autonomous and Adaptive Systems (NASA Autonomous Space Lander)
  - Causal Inference and Transfer Learning (ML Theory)
  - Adversarial Machine Learning (ATHENA, a defense framework for ML Systems)


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

I am primarily a software and systems researcher, but I am also interested in theory and robotics!

- **Computer Systems** (EuroSys, SoCC, JSys, TCC, FGCS)
- **Software Engineering** (ICSE, FSE, ASE, TSE, TOSEM, TAAS)
- **AI/ML** (UAI, AAMAS, AAAI, AutoML, JAIR)
- **Robotics** (IROS, RA-L, T-RO)

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

Sponsors:

Google

NSF NASA DARPA

Fatemeh Ghofrani
(PhD student)

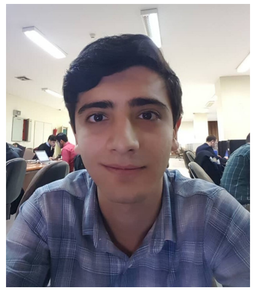
Abir Hossen
(PhD student)

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)

Collaborators:

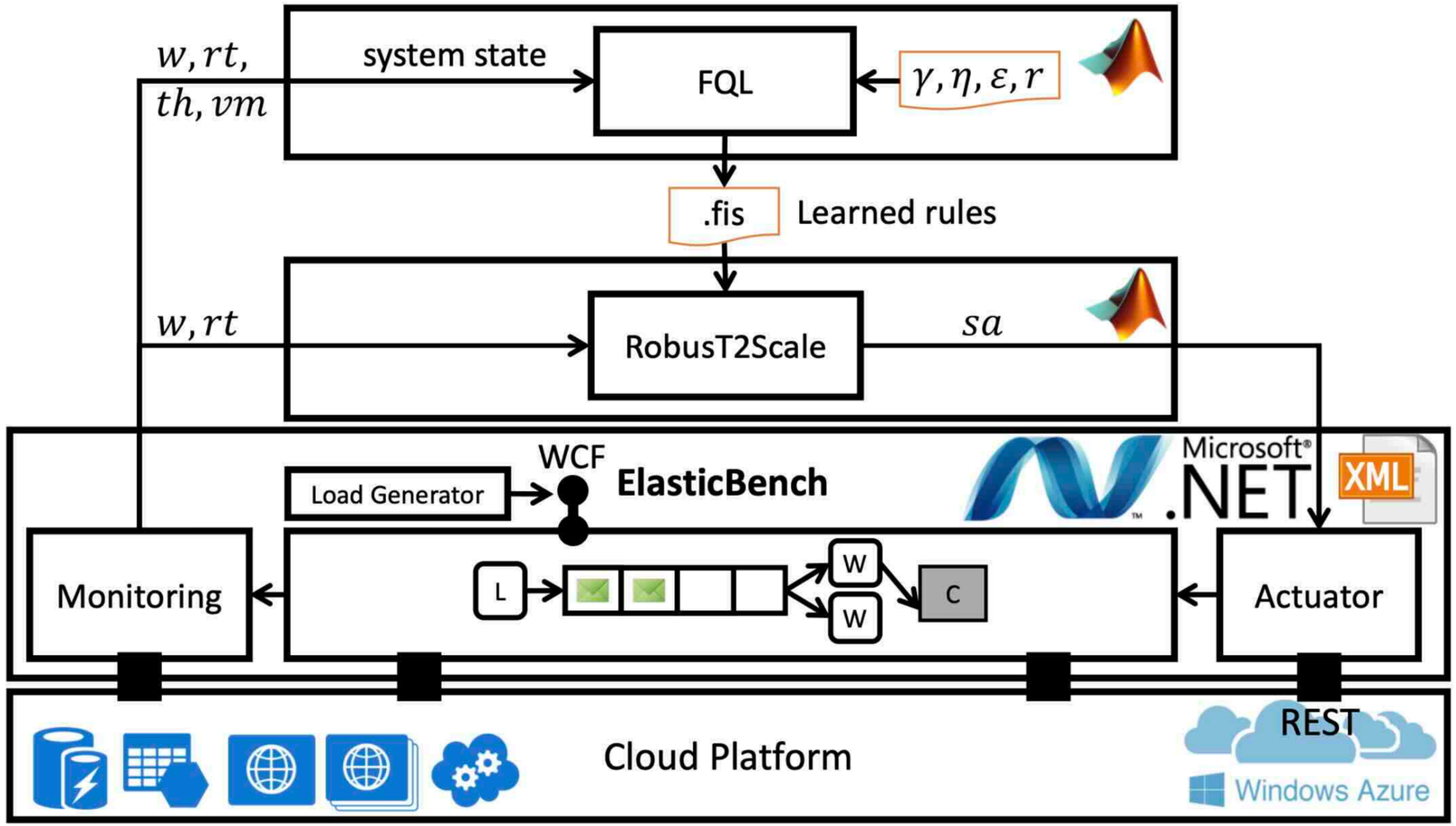
My story …

Machine Learning → Learning Systems

Back in 2010

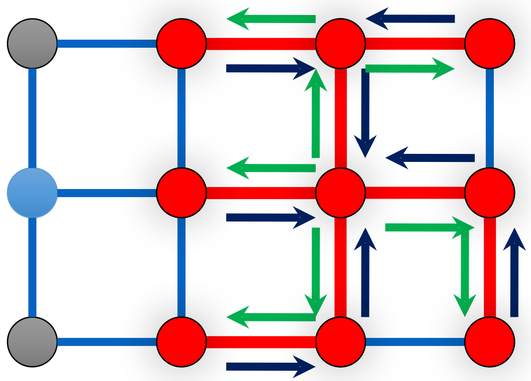
I started studying the use of Reinforcement Learning and Fuzzy Control for Cloud Auto-scaling.

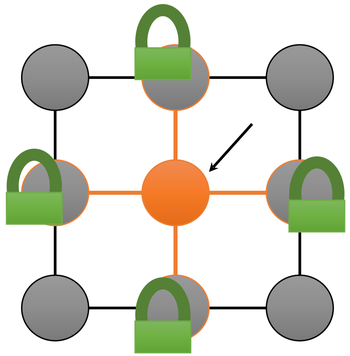
Research was slowed by the speed of RL training … and the fact that system non-functional qualities are difficult to predict



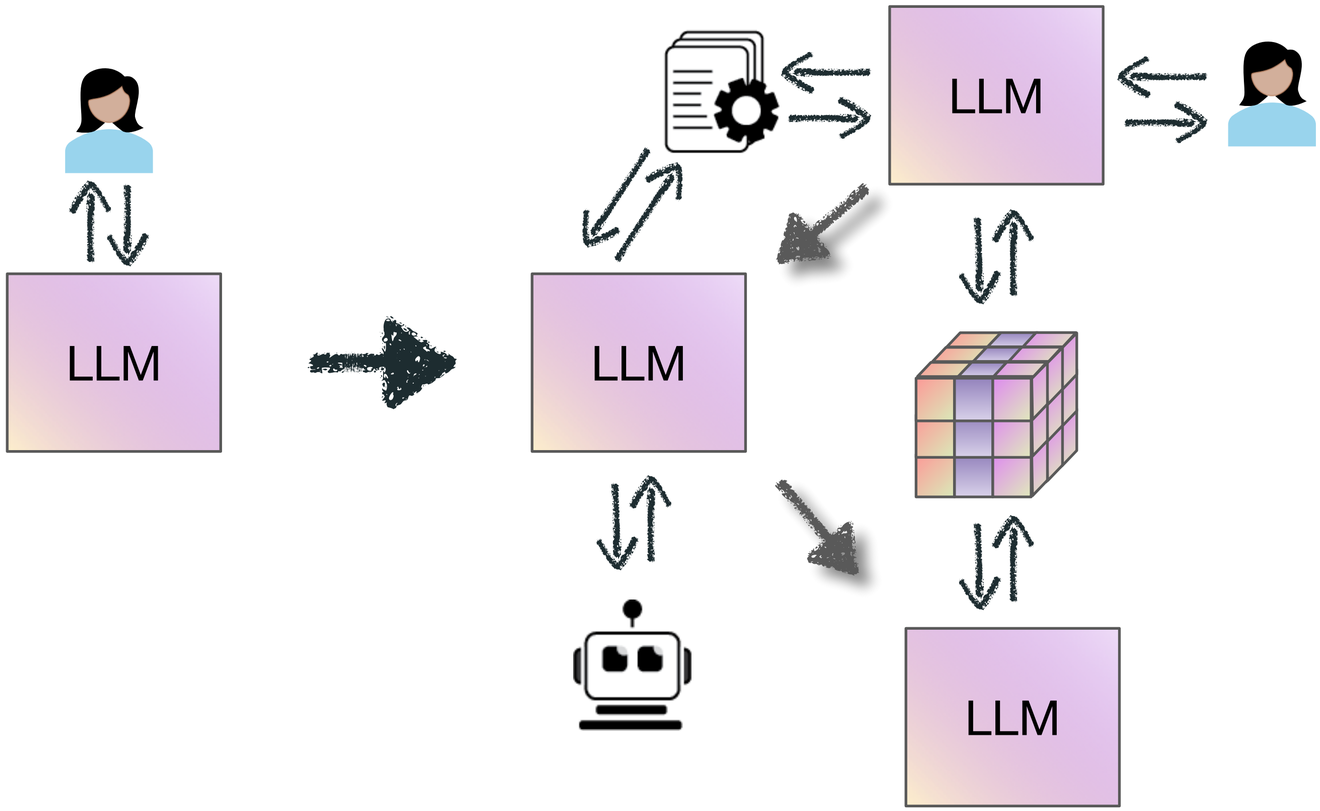
Changed Research Focus

I started studying parallel inference algorithms and systems



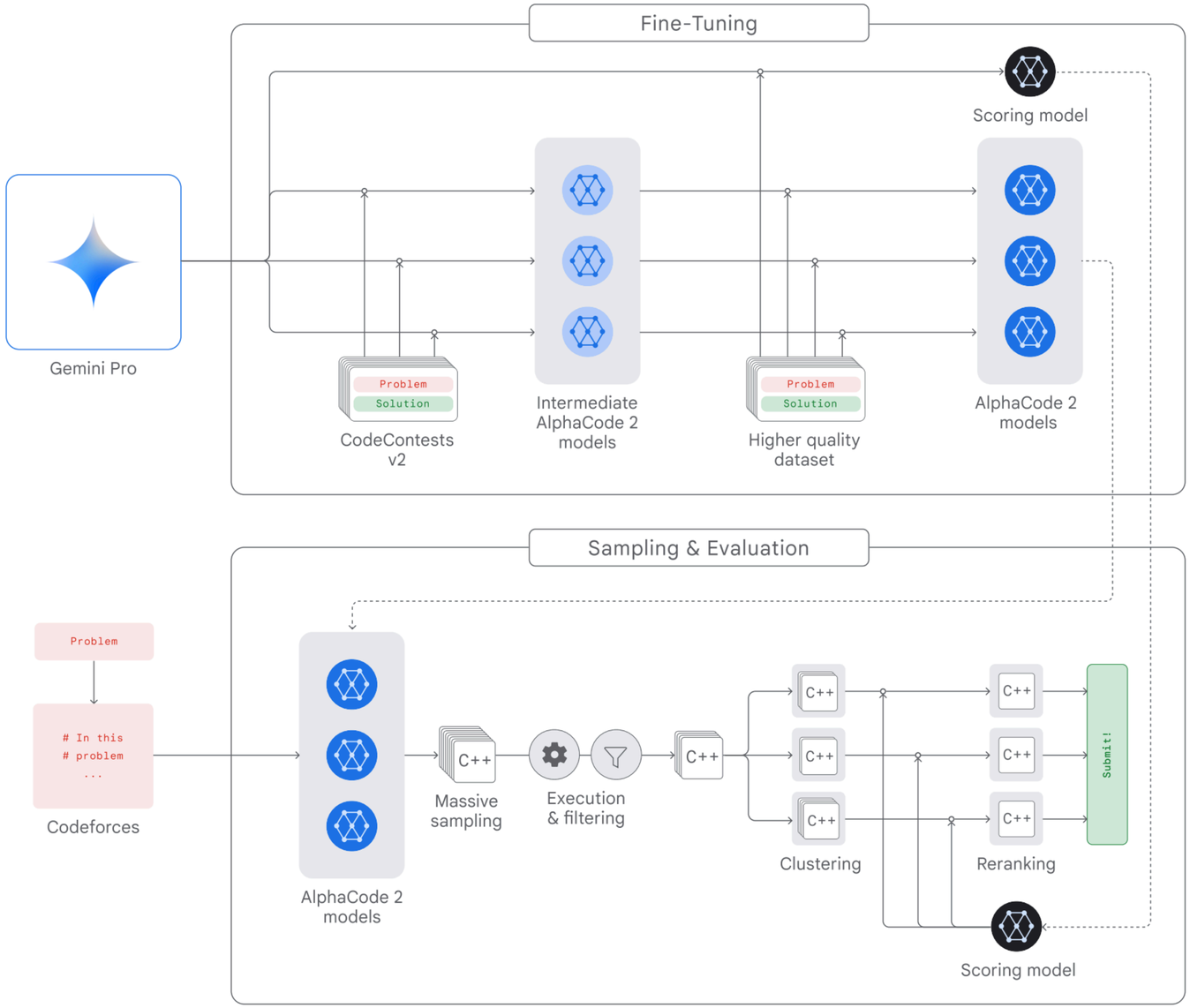


State-of-the-art AI results are increasingly obtained by systems composed of multiple components, not just monolithic models



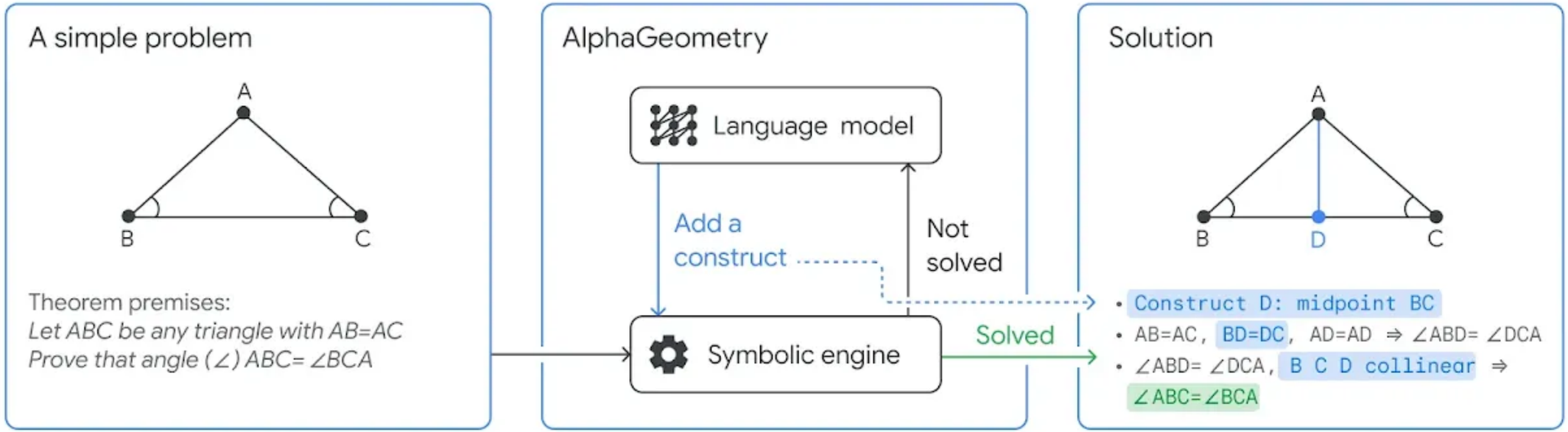
State-of-the-art AI results are increasingly obtained by systems composed of multiple components, not just monolithic models

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.



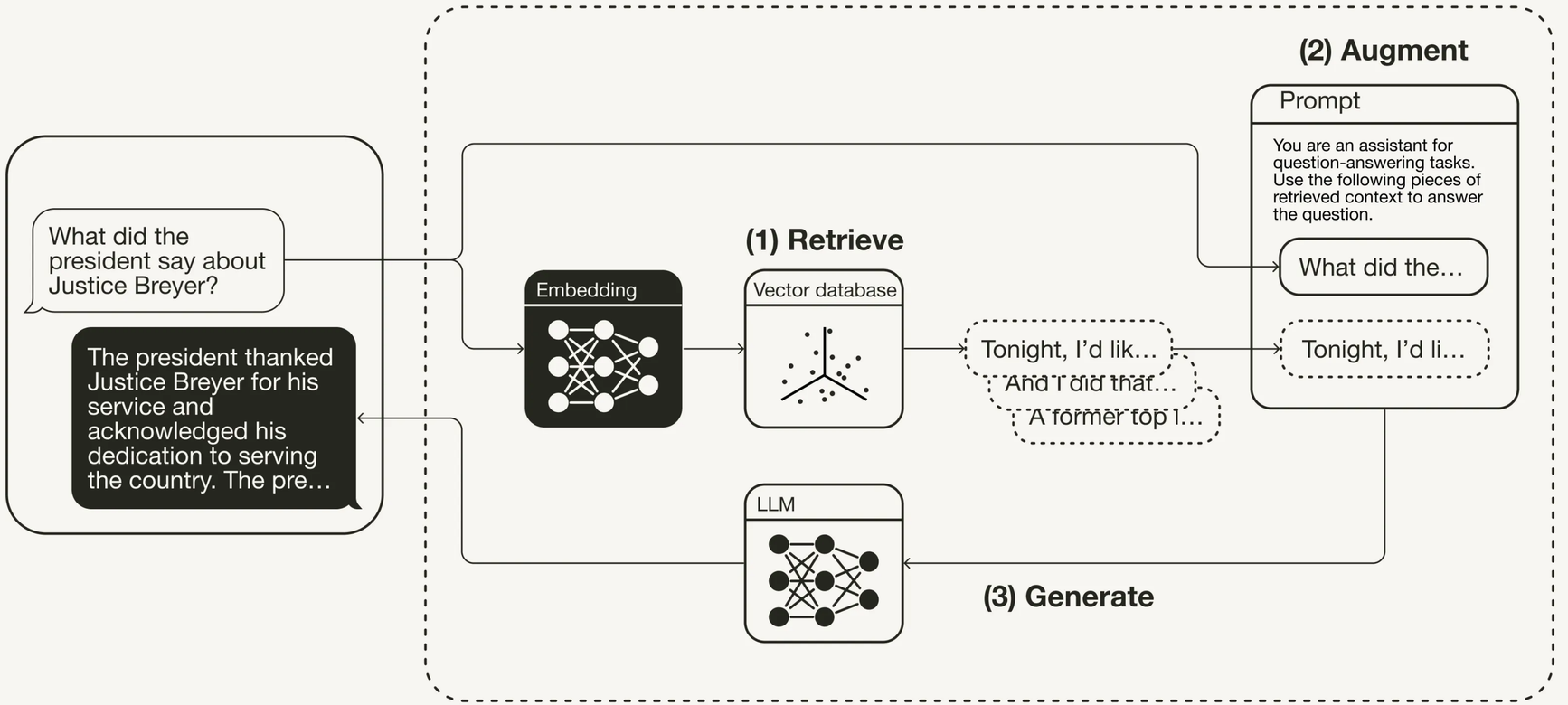
State-of-the-art AI results are increasingly obtained by systems composed of multiple components, not just monolithic models

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



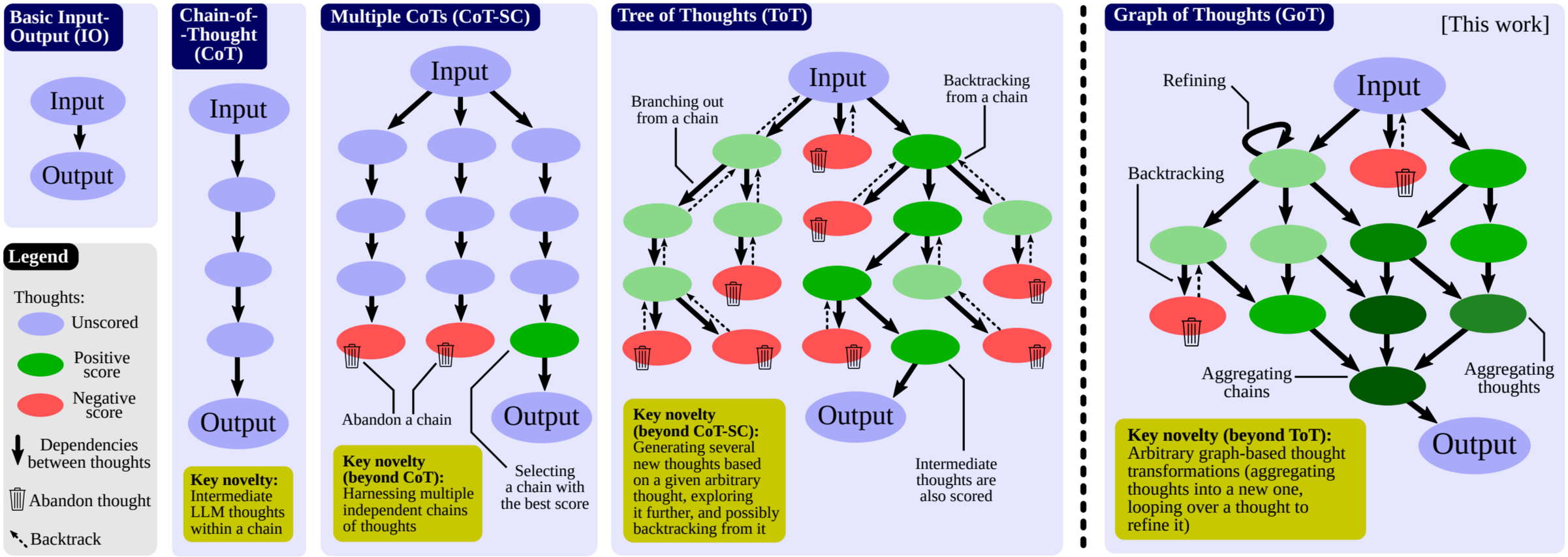
State-of-the-art AI results are increasingly obtained by systems composed of multiple components, not just monolithic models

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



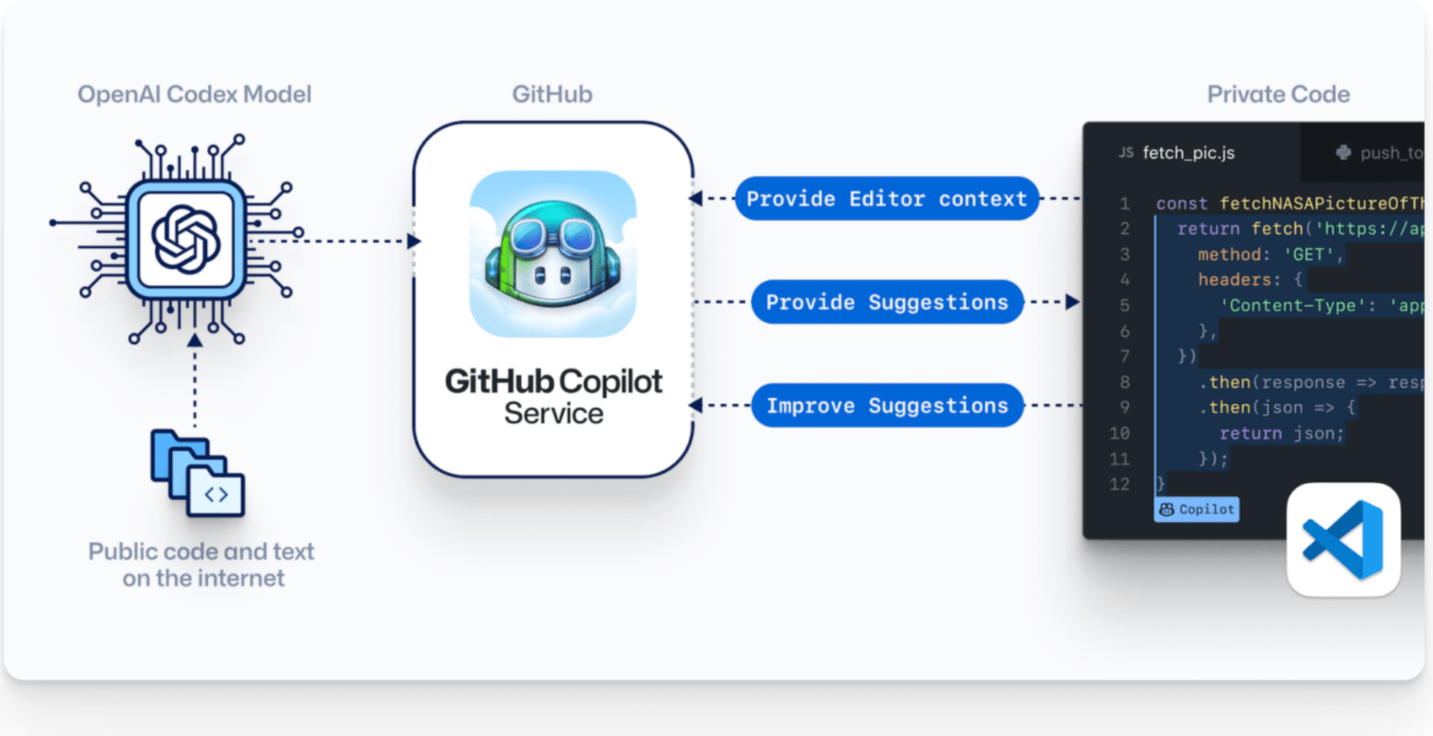
State-of-the-art AI results are increasingly obtained by systems composed of multiple components, not just monolithic models

…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.



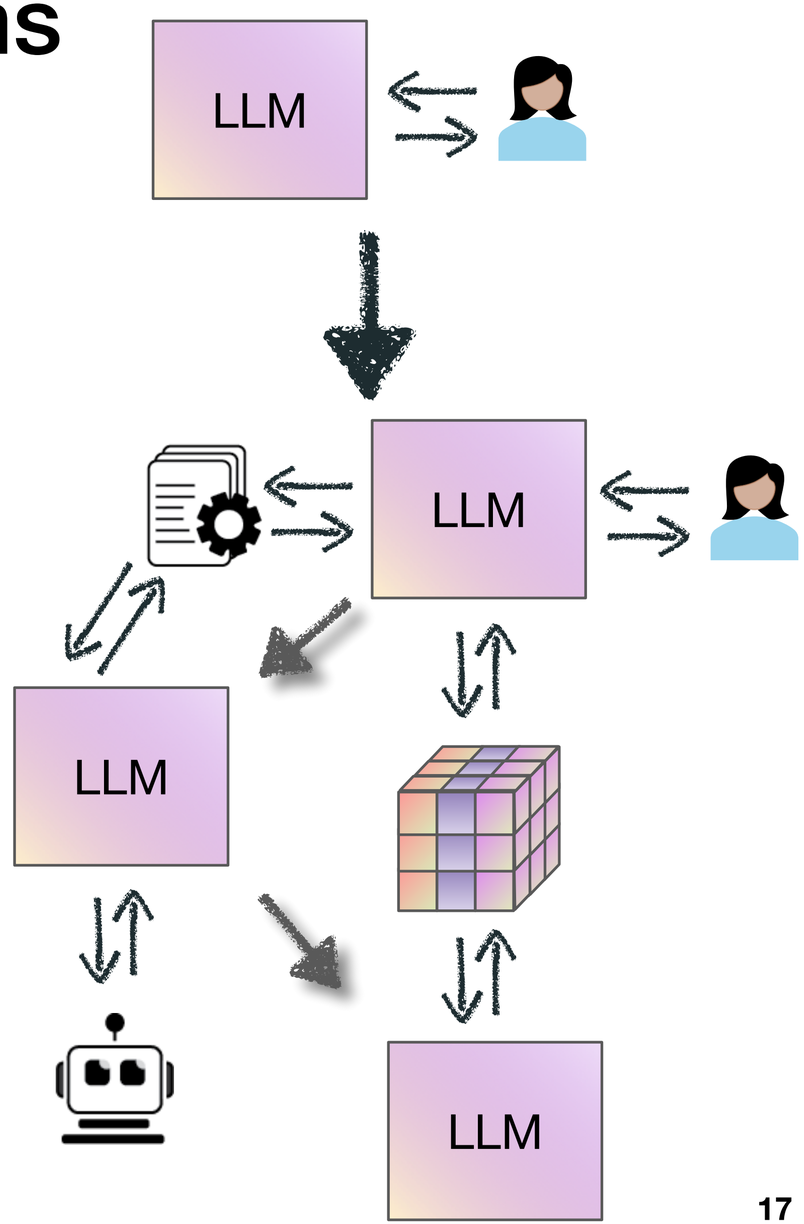
State-of-the-art AI results are increasingly obtained by systems composed of multiple components, not just monolithic models

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*	79.13% CoT@8*	87.29% CoT@32 (via API**)	70% 5-shot	78.4% 5-shot	78.5% 5-shot CoT	79.6% 5-shot	73.0% 5-shot	68.0%***
	83.7% 5-shot	71.8% 5-shot	86.4% 5-shot (reported)						

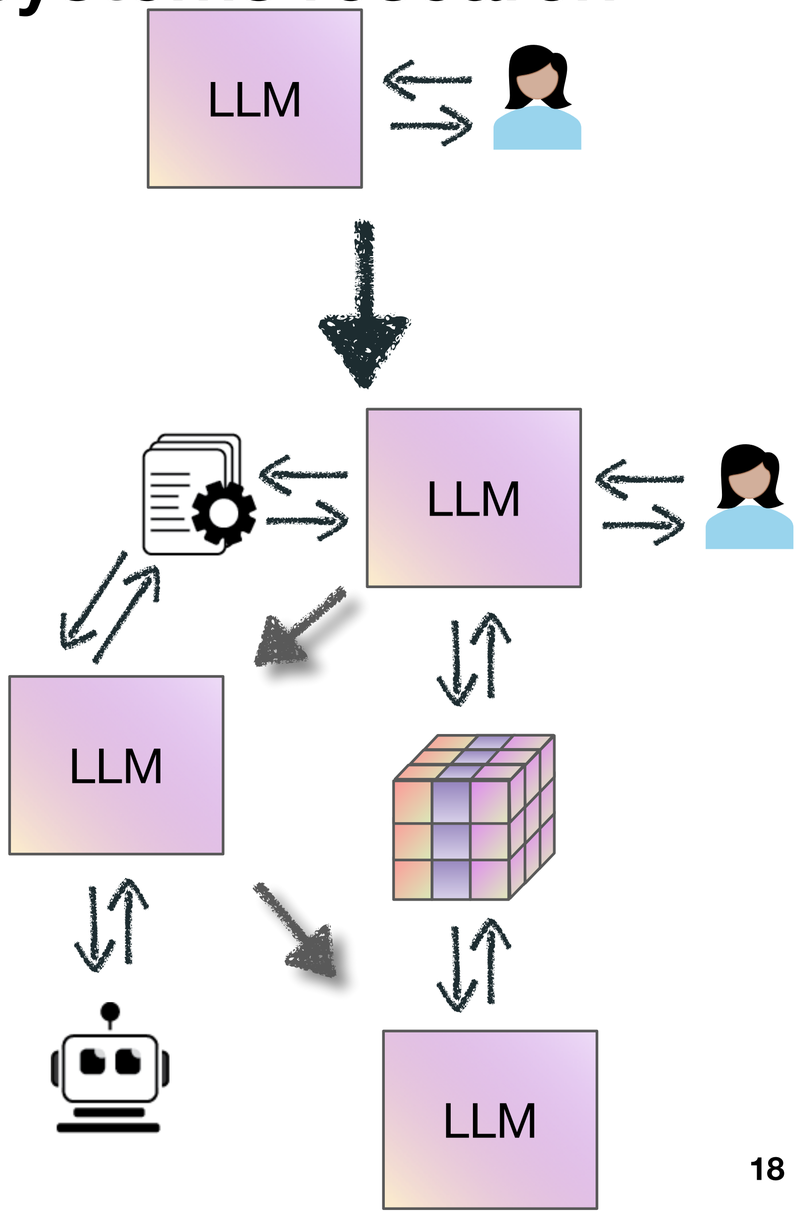
The paradigm shift from monolithic to modular-composed machine learning systems

- **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.



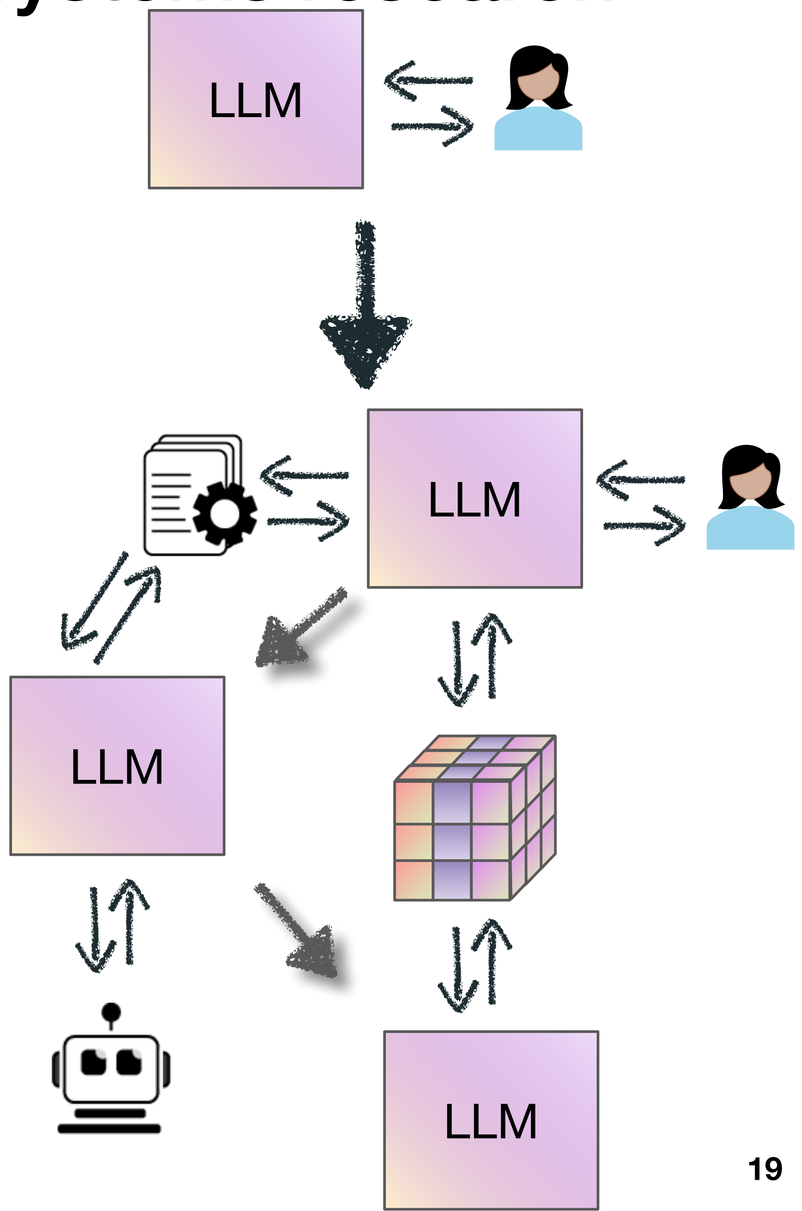
This paradigm shift to modular-composed ML systems opens up new opportunities for computer systems research

- 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.

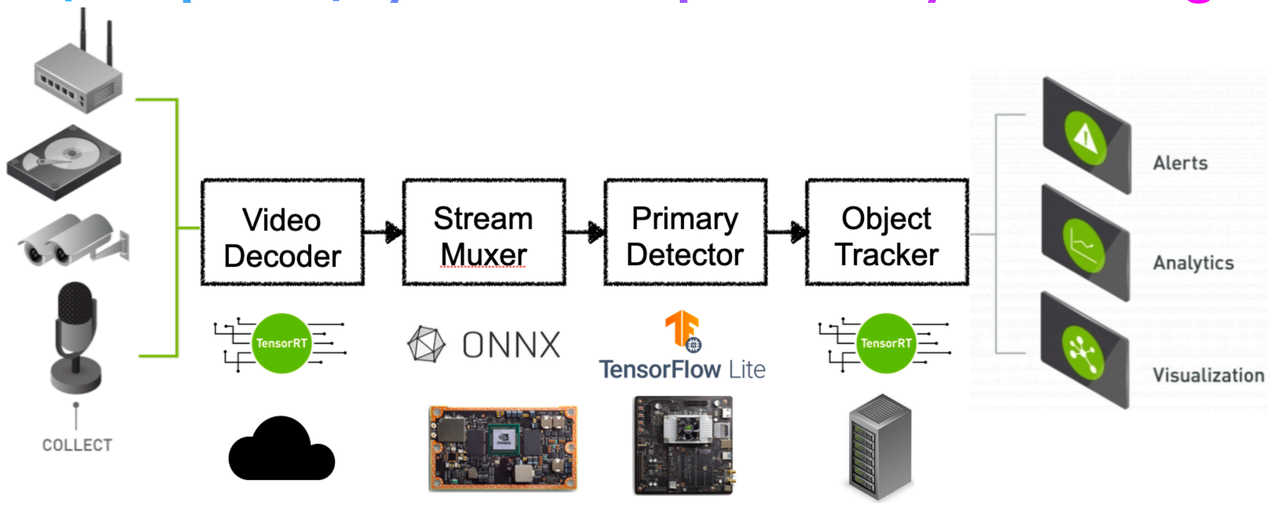


This paradigm shift to modular-composed ML systems opens up new opportunities for computer systems research

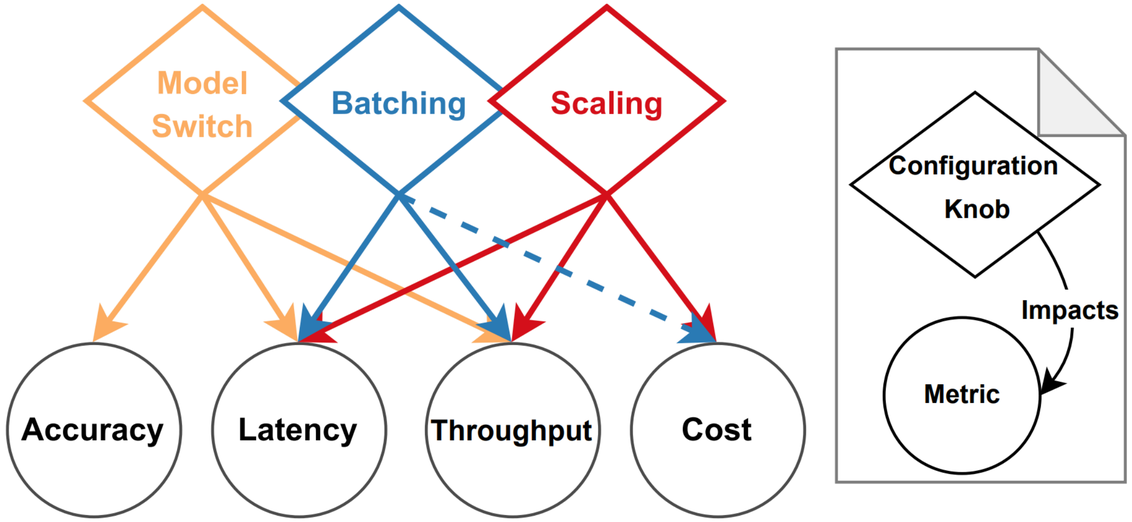
- This shift to modular-composed systems opens many **interesting systems questions.**
- It is also exciting because it means leading AI results can be achieved through clever **systems ideas**, not just scaling up training.



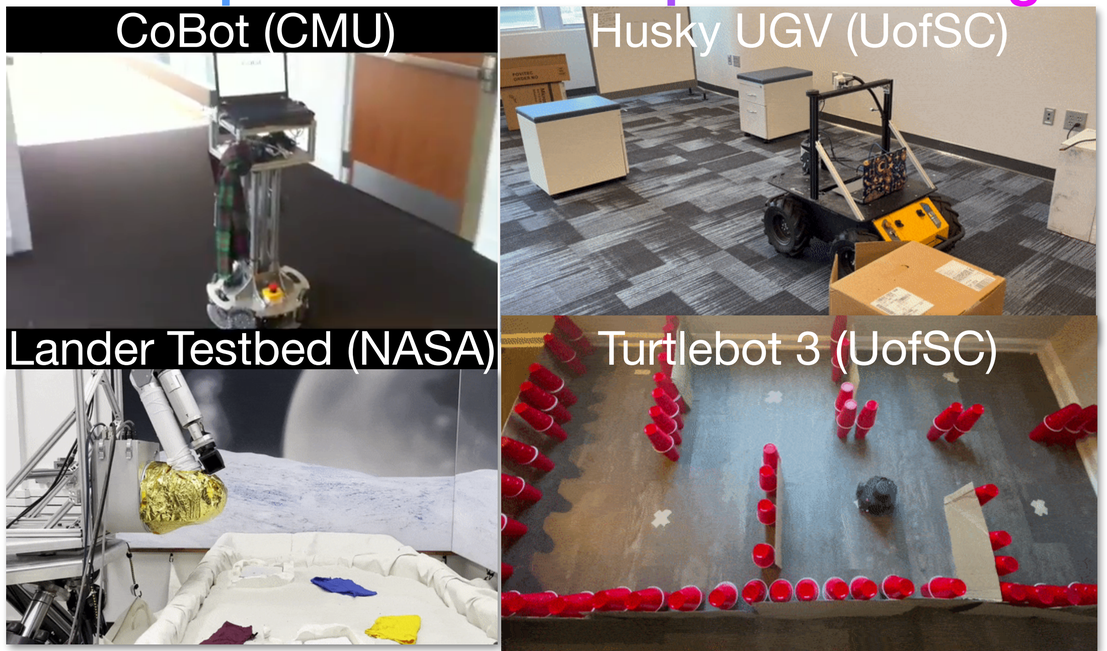
The variability space (design space) of (composed) systems is exponentially increasing



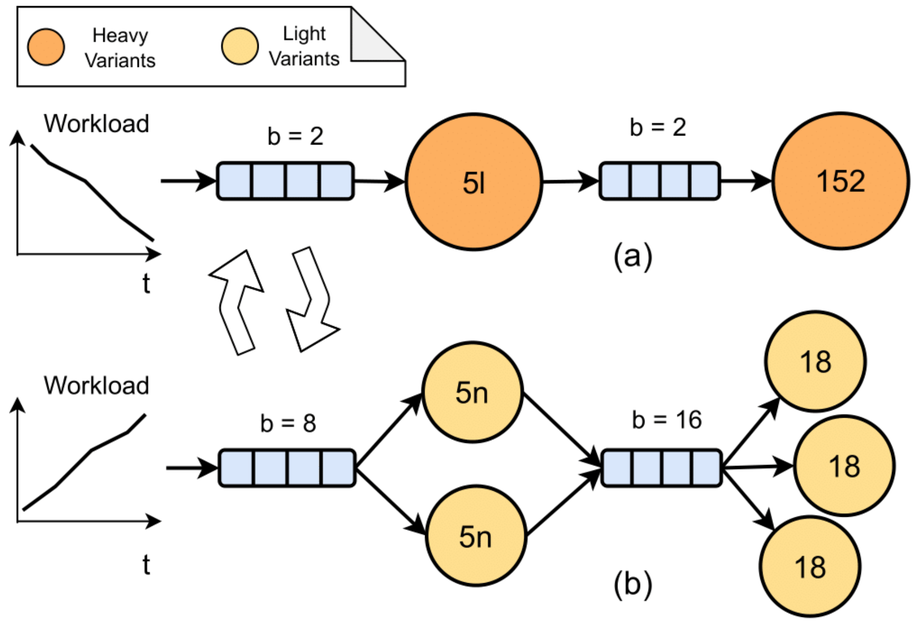
Performance goals are competing and users have preferences over these goals



Systems operate in uncertain environments with imperfect and incomplete knowledge

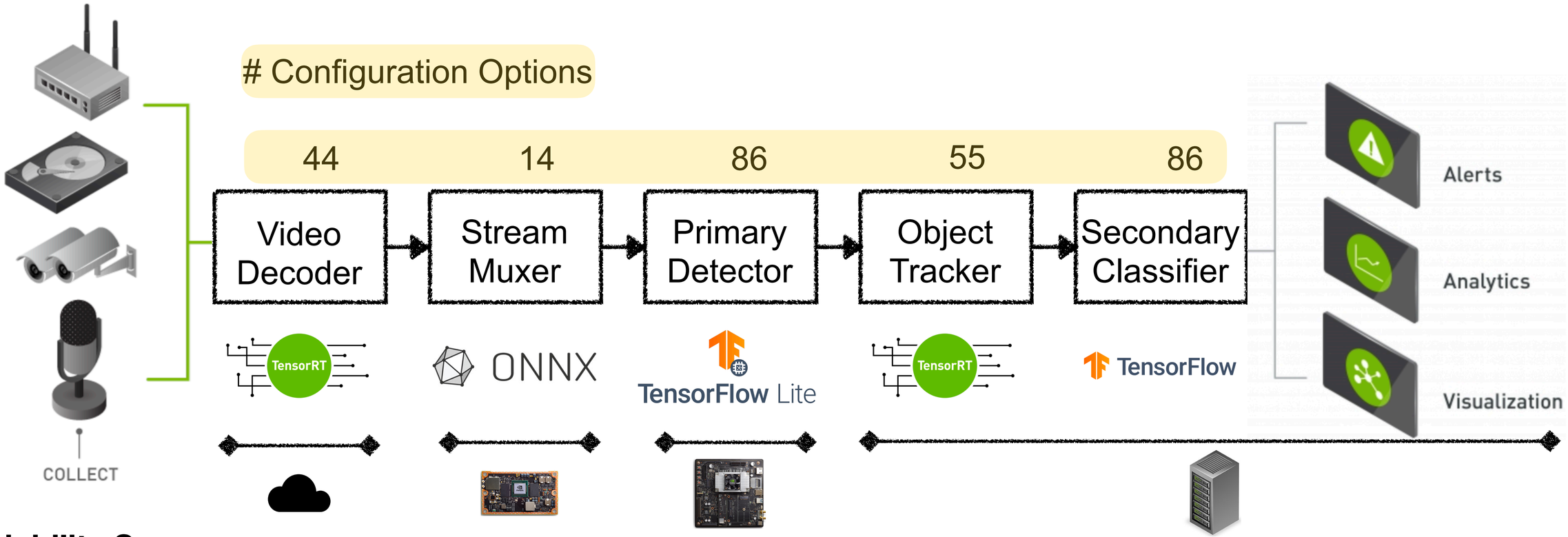


Goal: Enabling users to find the right quality tradeoff

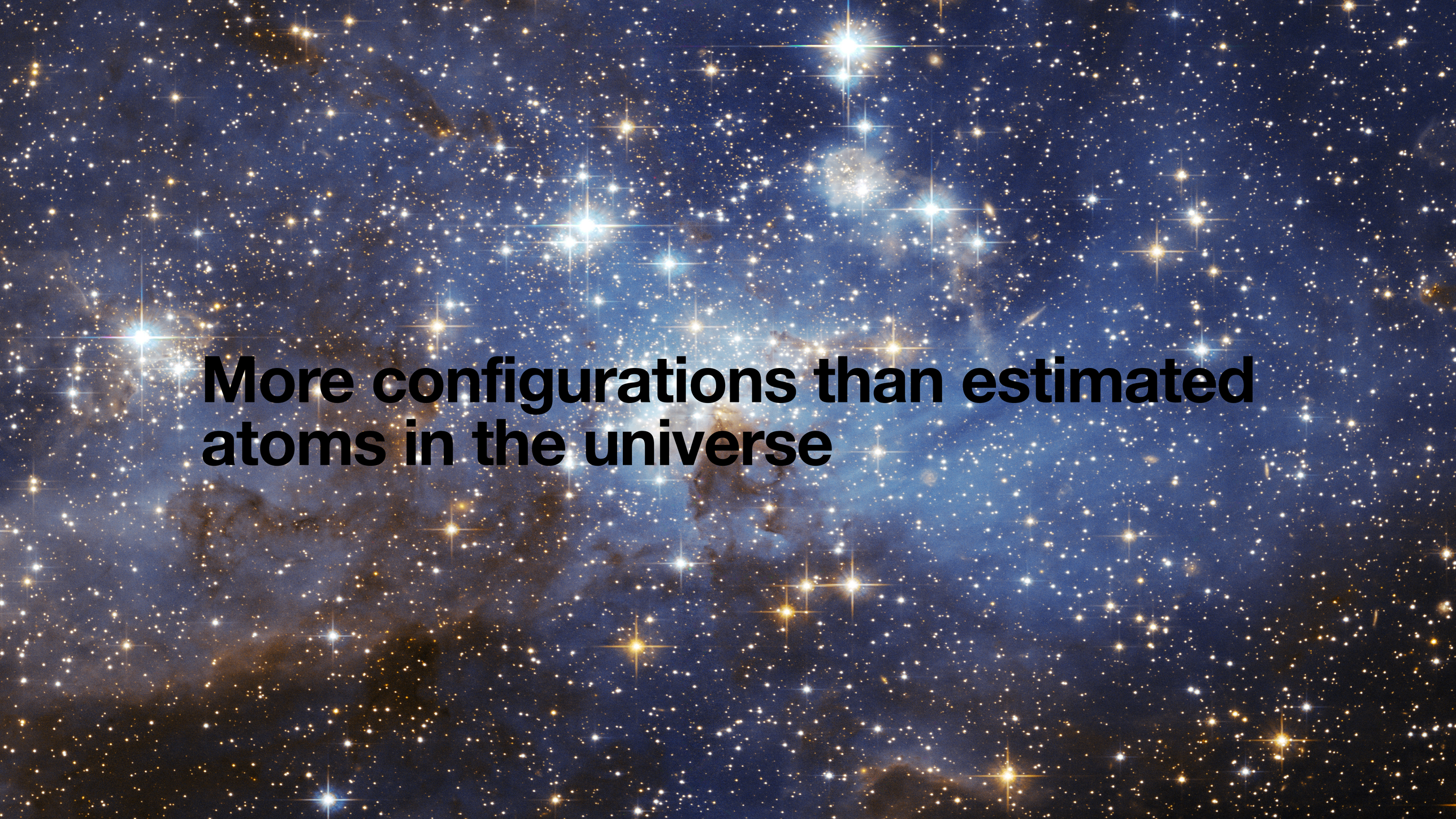


The variability space of today's systems is exponentially increasing

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

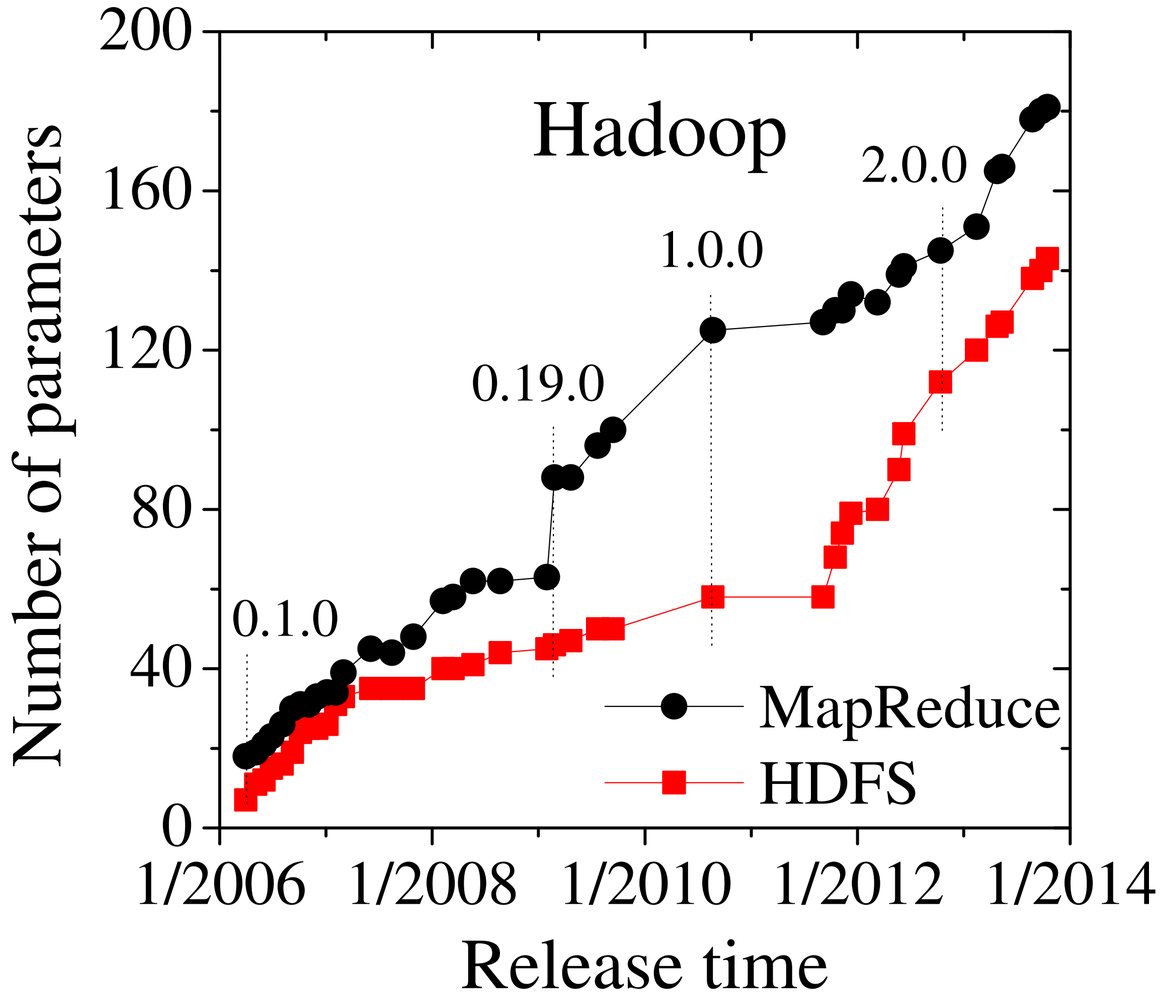


Variability Space =
Algorithm Selection +
Configuration Space +
System Architecture +
Deployment Environment


More configurations than estimated
atoms in the universe

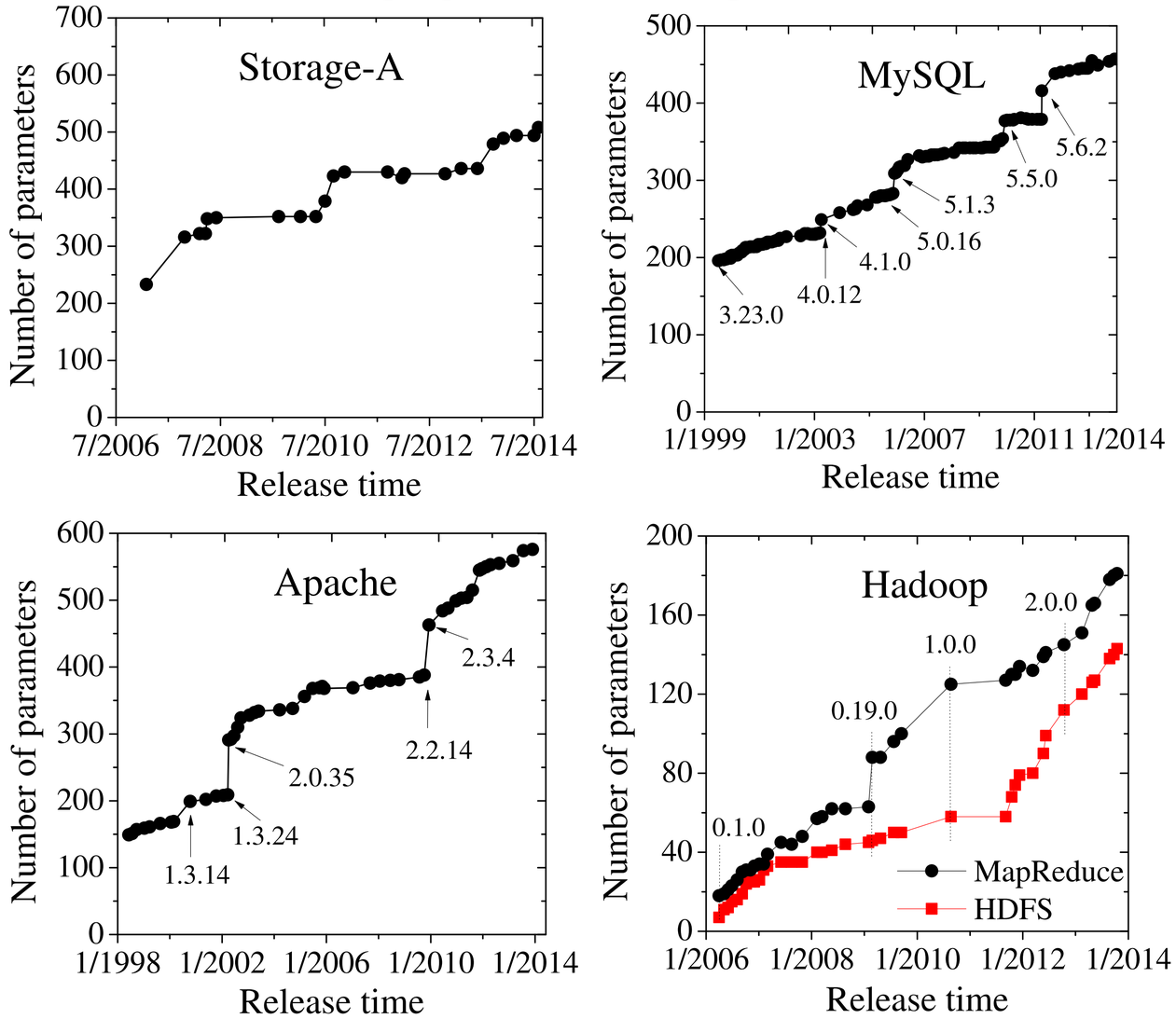
```
102
103 drpc.port: 3772
104 drpc.worker.threads: 64
105 drpc.max_buffer_size: 1048576
106 drpc.queue.size: 128
107 drpc.invocations.port: 3773
108 drpc.invocations.threads: 64
109 drpc.request.timeout.secs: 600
110 drpc.childopts: "-Xmx768m"
111 drpc.http.port: 3774
112 drpc.https.port: -1
113 drpc.https.keystore.password: ""
114 drpc.https.keystore.type: "JKS"
115 drpc.http.creds.plugin: org.apache.storm.security.auth.DefaultHttpCredentialsPlugi
116 drpc.authorizer.acl.filename: "drpc-auth-acl.yaml"
117 drpc.authorizer.acl.strict: false
118
119 transactional.zookeeper.root: "/transactional"
120 transactional.zookeeper.servers: null
121 transactional.zookeeper.port: null
122
123 ## blobstore configs
124 supervisor.blobstore.class: "org.apache.storm.blobstore.NimbusBlobStore"
125 supervisor.blobstore.download.thread.count: 5
126 supervisor.blobstore.download.max_retries: 3
127 supervisor.localizer.cache.target.size.mb: 10240
128 supervisor.localizer.cleanup.interval.ms: 600000
129
```

Empirical observations confirm that systems are becoming increasingly configurable



[Tianyin Xu, et al., "**Too Many Knobs**…", FSE'15]

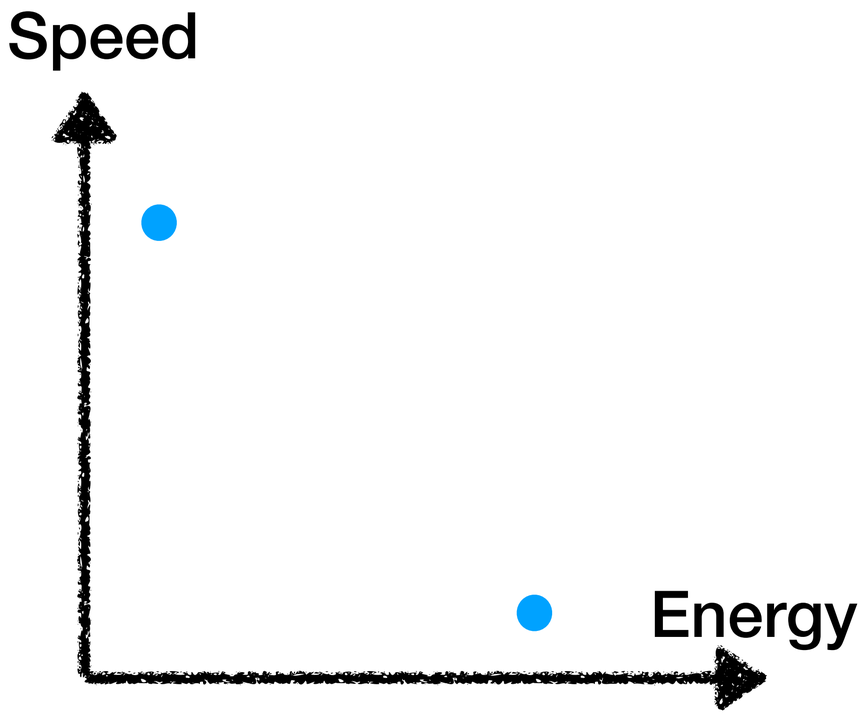
Empirical observations confirm that systems are becoming increasingly configurable



[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
}
```



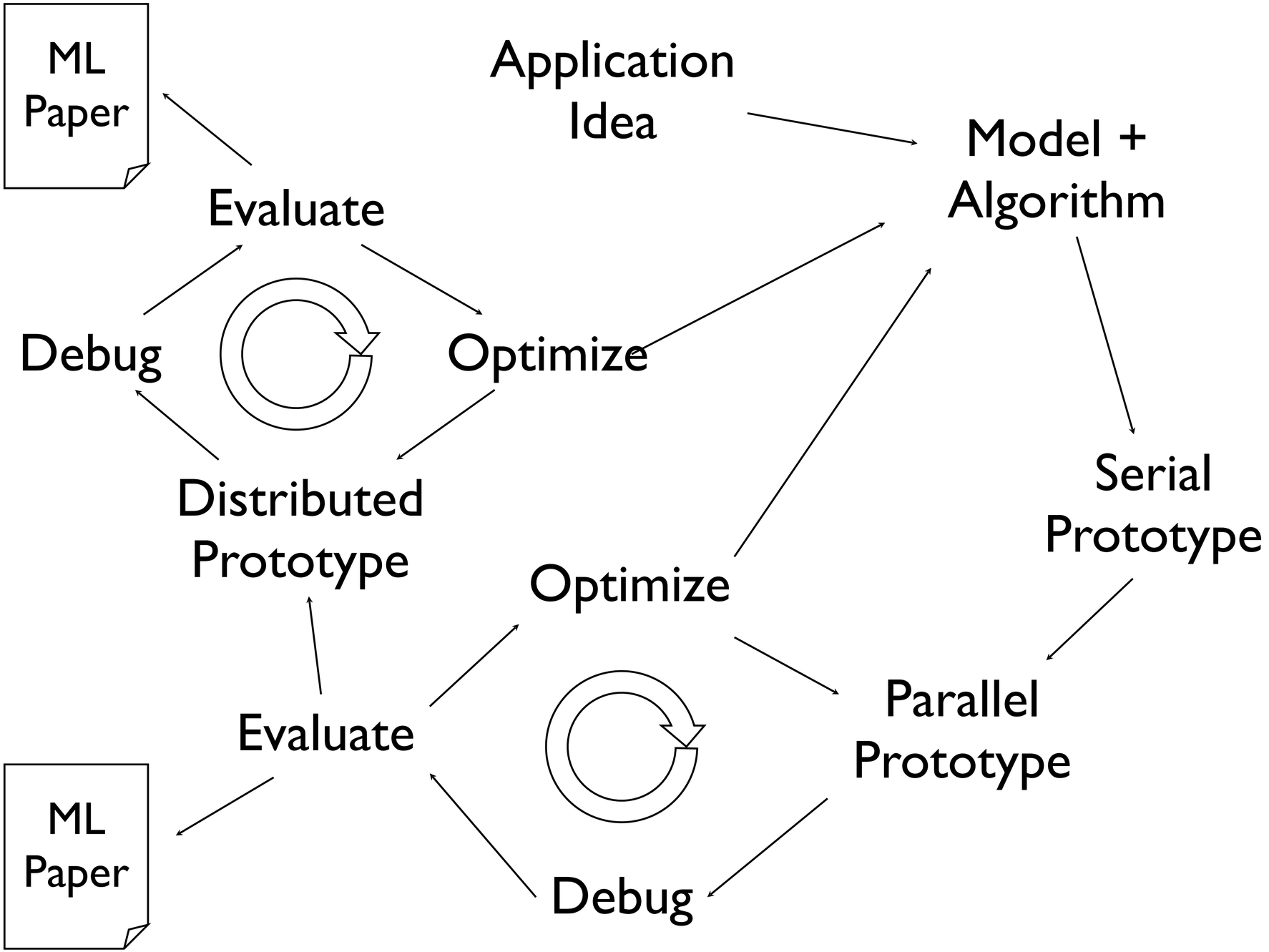
Challenges of configurations

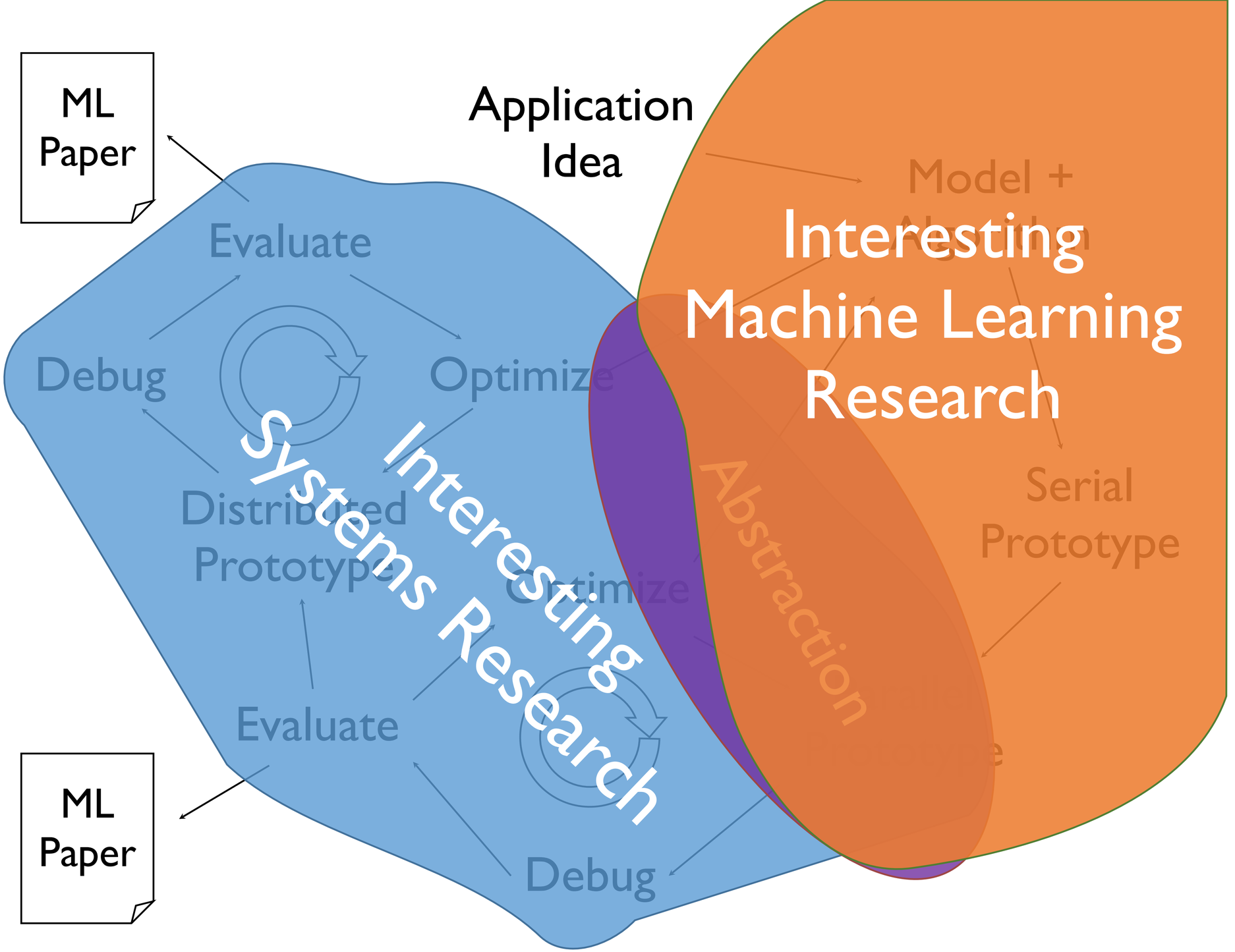
- Difficulties in knowing which parameters should be set
- Setting the parameters to obtain the intended behavior
- Reasoning about multiple objectives (energy, speed)

The goal of my research is…

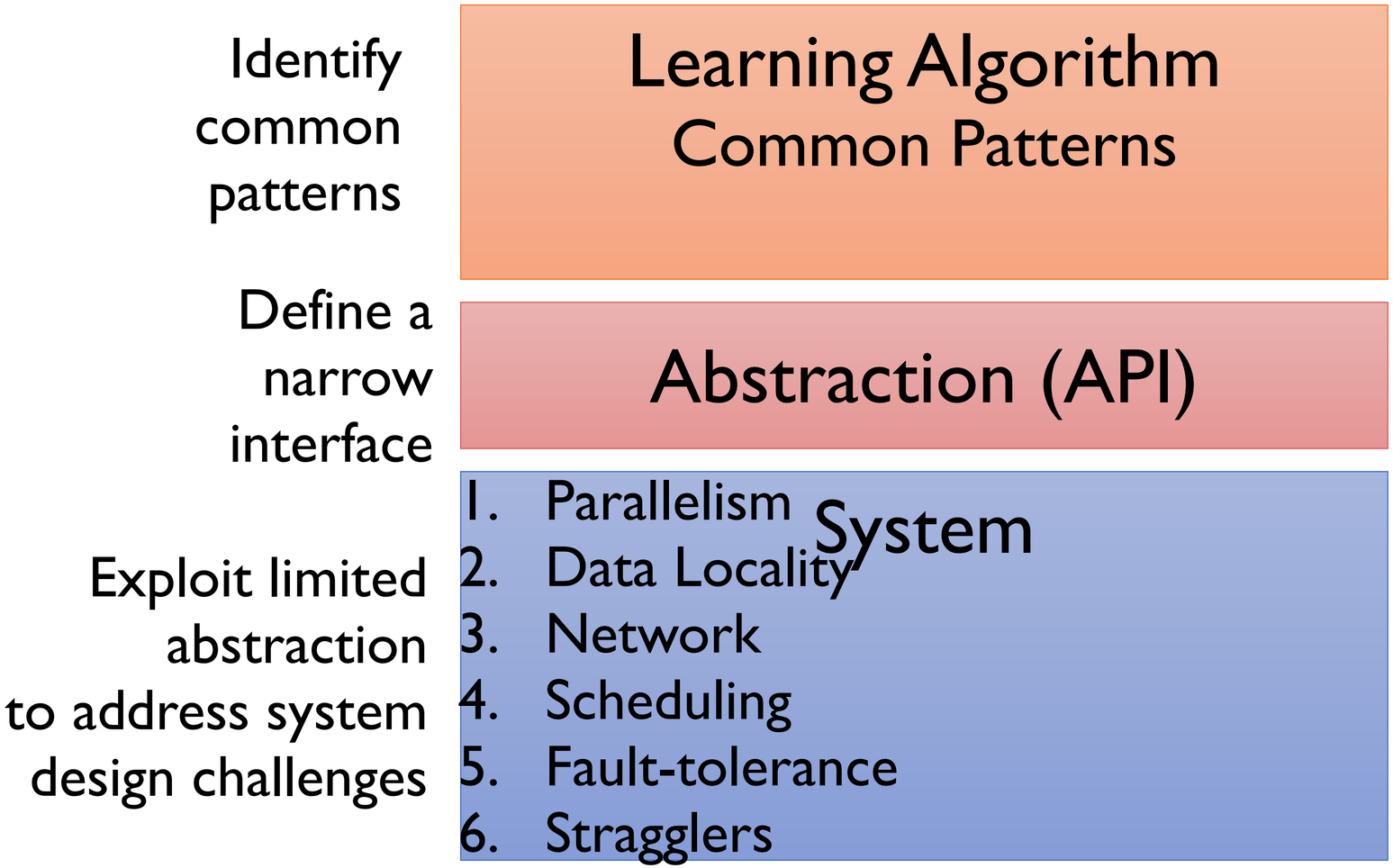
Understanding the performance behavior of real-world highly-configurable systems that scale well…

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


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

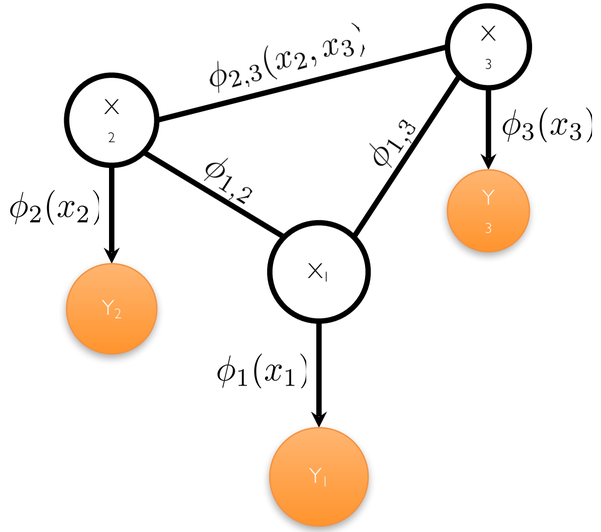

ML Paper
Application Idea
Evaluate
Debug
Optimize
Distributed Prototype
Evaluate
Debug
ML Paper
Interesting Systems Research
Model +
Serial Prototype
Optimize
Abstraction
Interesting Machine Learning Research

Managing Complexity Through Abstraction

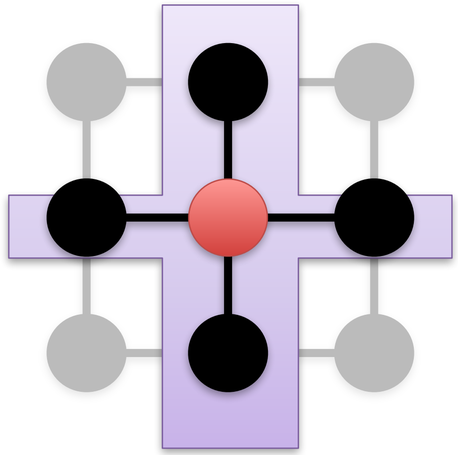


PhD in Machine Learning from CMU 2013

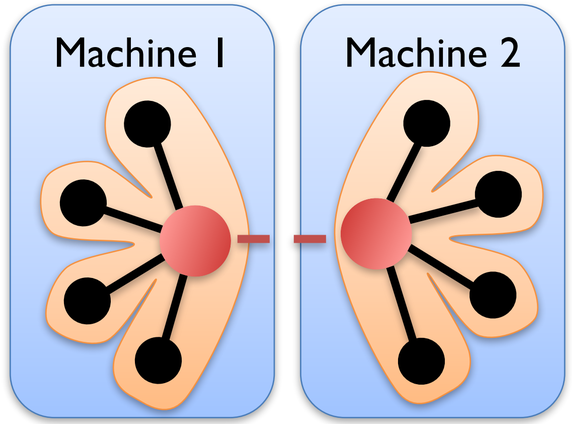
Machine Learning | Abstractions | Scalable Systems



Graphical Model Inference



Vertex Program



GraphLab/ GraphX System

What do you expect to learn in this course?

Course Overview

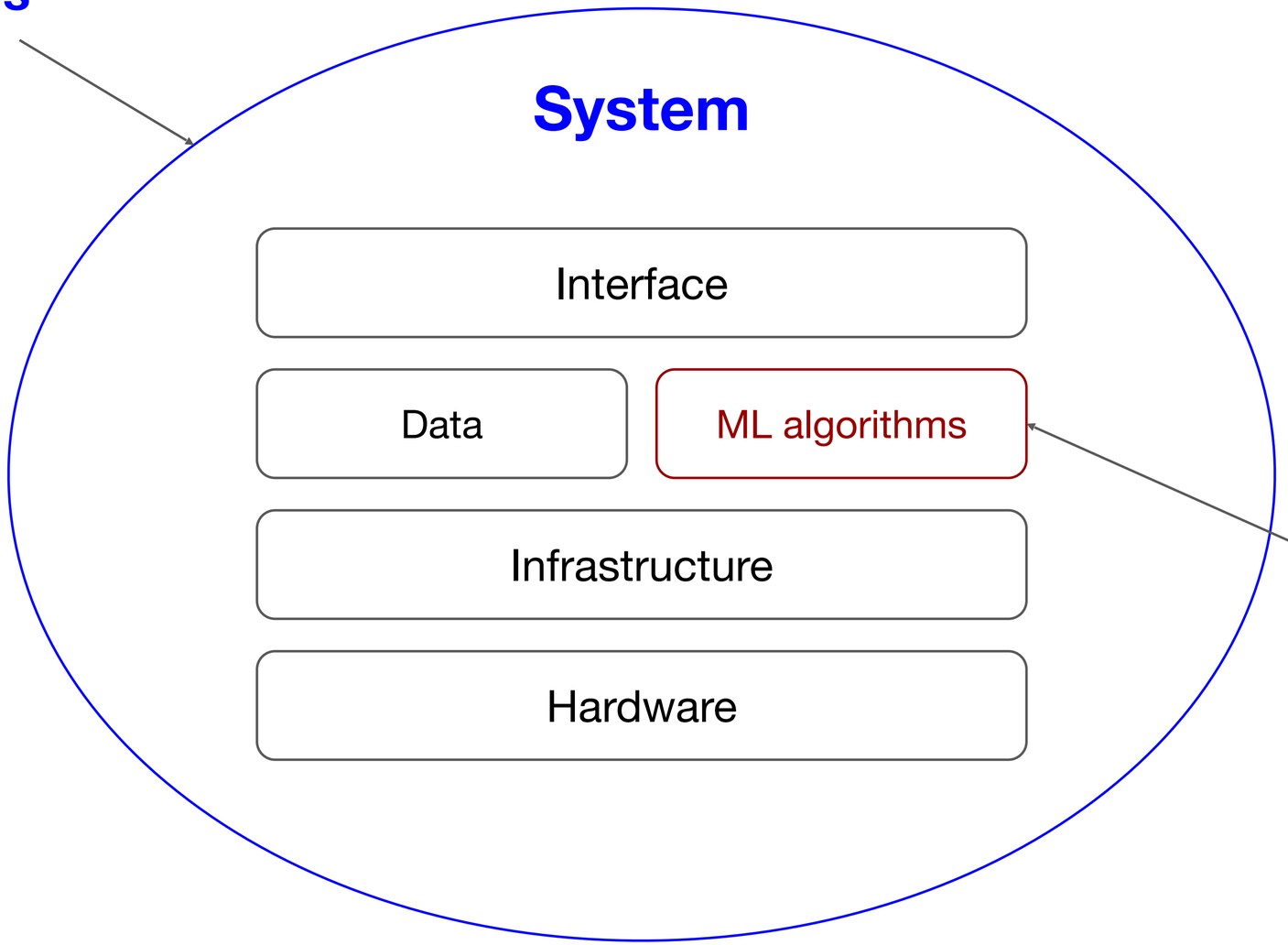
Why ML Systems instead of ML algorithms?

- 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?

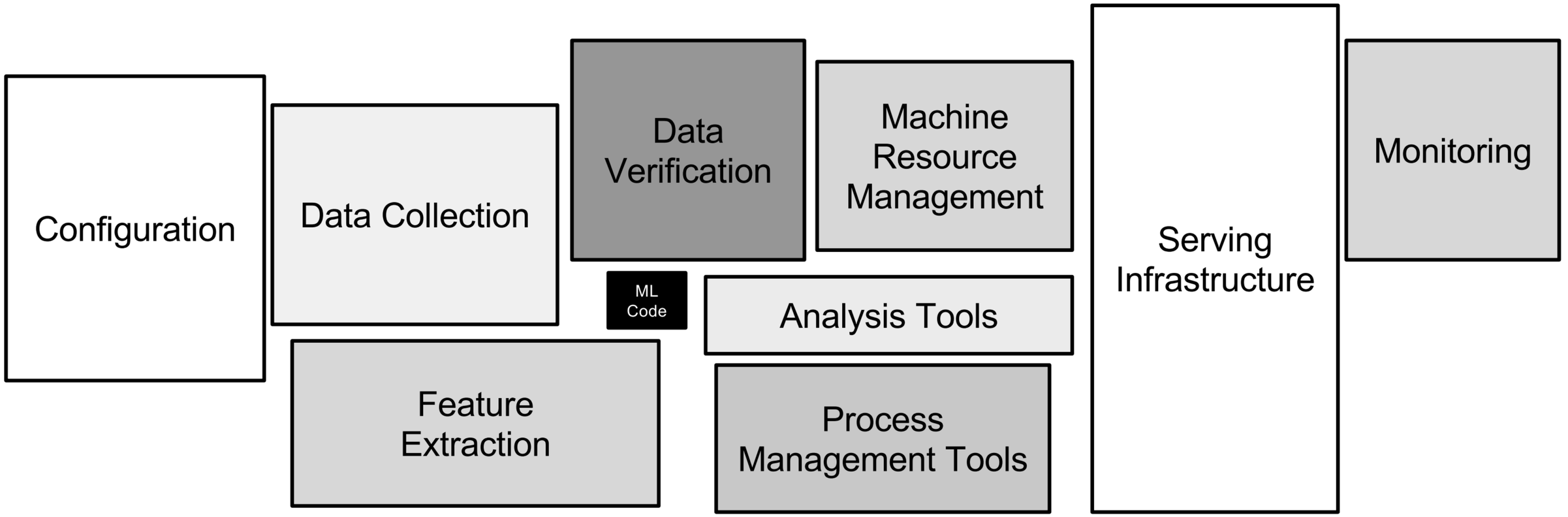
- ML algorithms is the less problematic part.
- The hard part is to **how to make algorithms work with other parts to solve real-world problems**.
- 60/96 failures caused by non-ML components

CSCE 585:
ML Systems



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

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!

Evaluations

ML Systems course is project-based

- Must work in groups of 2-3
- Demo + report + code + experimental results

See course website: https://pooyanjamshidi.github.io/mls/

Important Dates?

Important Dates

- Project assignments and proposals: due September 5
- Project milestone 1: due September 28
- Project milestone 2: due October 26
- Project demos: November 28th, 30th, Dec 5th, Dec 7th
- Final report and all deliverables: due December 7th

Piazza

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

How projects will be evaluated

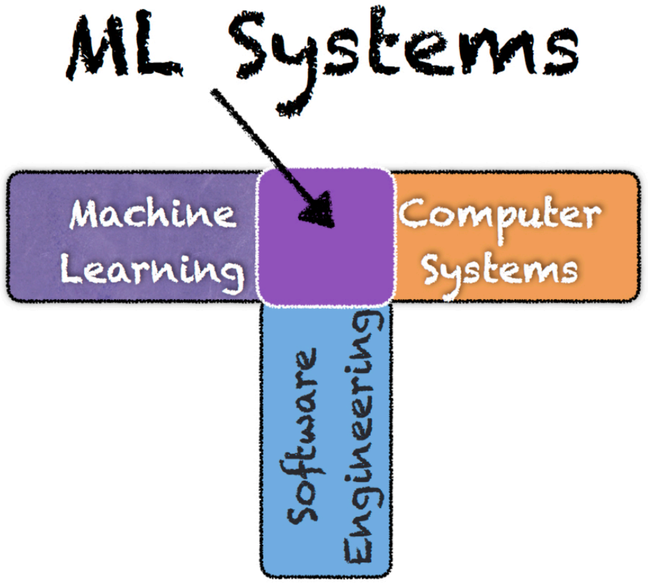
- You can work in teams of up to 2 or 3 people.
- Every team member should be able to demonstrate her/his contribution(s).
- The outcome will be evaluated based on the quality of the deliverables (code, results, report) and presentations/demonstrations.
- The final report contains motivation, positioning 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



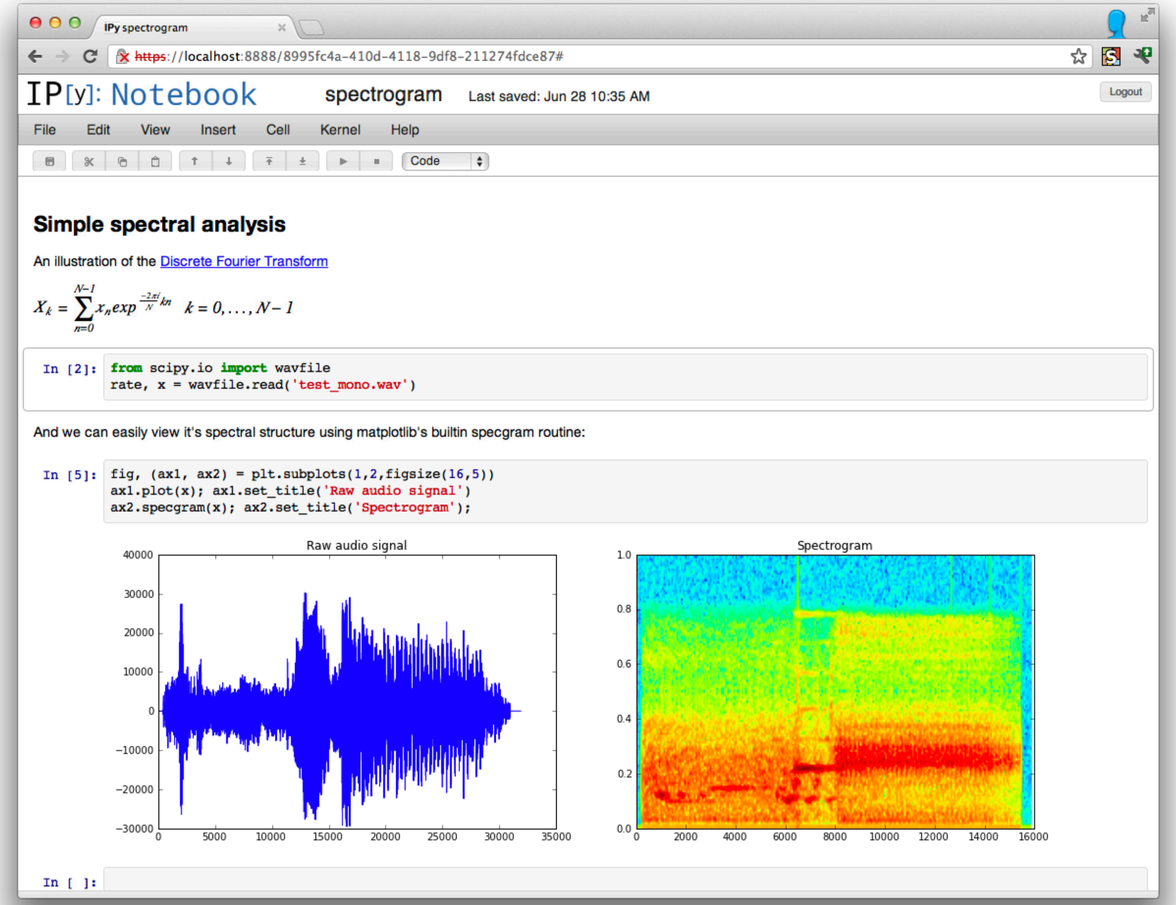
New to machine learning? Not sure how ML works in production? Interested to get involved in advanced ML+Systems research? This class is designed for you!

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.

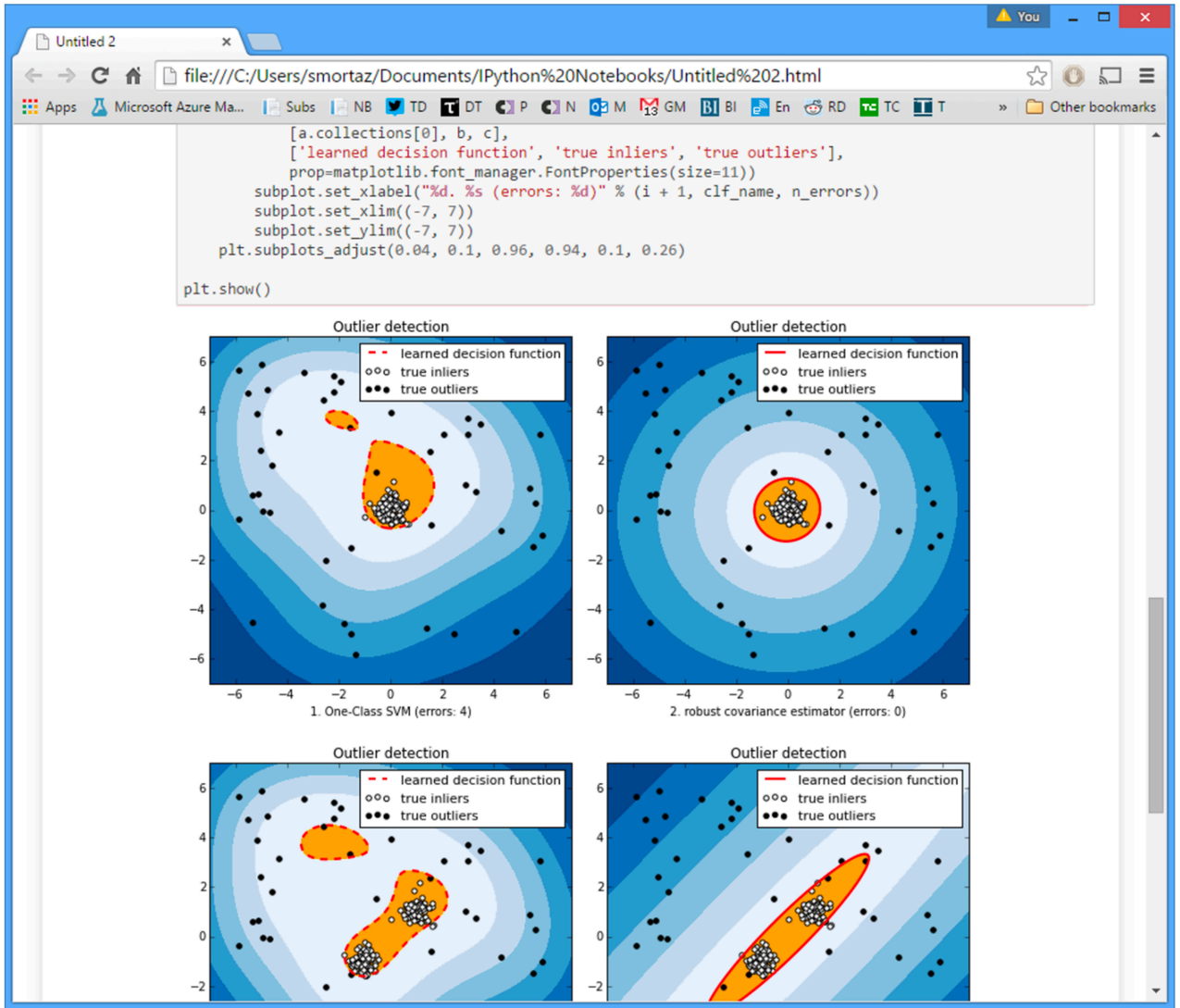
https://pooyanjamshidi.github.io/mls/

Examples, Tips, Suggestions

How the project report should looks like?



How the project report should looks like?



How the project report should looks like?

IP[y]: Notebook GDP_CO2_Example Last saved: Feb 26 12:33 PM

File Edit View Insert Cell Kernel Help

Code

Andy Wilson has a nice more tightly intergration of d3.js and ipython notebook. See https://github.com/wilsaj/ipython/tree/d3plots

some other examples (mostly experimental, need various different setups.)

https://github.com/csohn/IPython-Notebook---d3.js-mashup

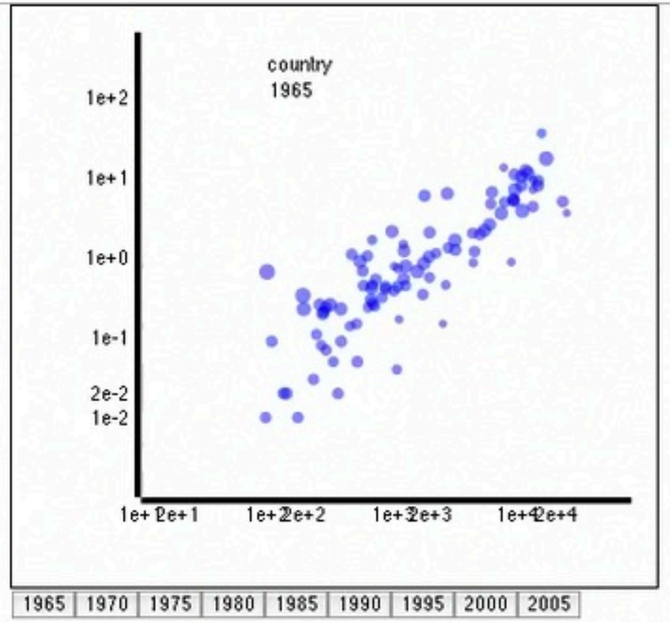
Why?

The whole exercise here is mostly on exploring the possibility to have really dynamic frontend for developing visulizations or demostrations. The ipython notebook provides a really nice way to integrate web technologies with the powerful backend python processes. This will make dynamic data exploratory work with python easier in the furture using mostly open-source software. We can eventually integrate a lots of other cool web technologies (e.g. webGL, html5 video, canvas) together.

What's next

In this example, I use bare-bone python functions / javascript functions for the work. I think the reasonable next step is to see what is the right kind of framework for mapping the javascript objects and python objects (e.g. something like https://github.com/mikedewar/d3py for ipython notebook or Andy Wilson's d3plots approach.) Eventually, we may develope a standard set of widgets or integrate some concept of the "Grammar of Graphics" (http://www.amazon.com/Grammar-Graphics-Leland-Wilkinson/dp/0387987746) and ggplot2-like features (http://had.co.nz/ggplot2/) as python notebook libraries.

--Jason Chin, Feb 26, 2012



In [35]:
```
# Here we show we can re-define the function and have the javascript calls
# the re-defined function immediately
# The code below plots the circles using the sizes propotional to the log o
# population of each country
```

ML Systems Projects

Course Projects

- **Project 1: ML Pipeline Adaptation**: Configuration Tuning, Runtime Resource Adaptation; Extensions over IPA (https://github.com/reconfigurable-ml-pipeline/ipa)
- **Project 2: You define the scope!** Any Relevant Topic to ML Systems;
- Examples:
  - **LLM Serving**: Batching, Scheduling, Eviction Policy, Prefetching
  - **Robotics:** Multi-Modal Learning and Navigation

Project Proposal

- What is the **problem** that you will be investigating? Why is it interesting?
- What **reading** will you examine to provide context and background?
- What **data** will you use? If you are collecting new data, how will you do it?
- What **method** or algorithm are you proposing? If there are existing implementations, will you use them and how? 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.
- 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)?

Checkout

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

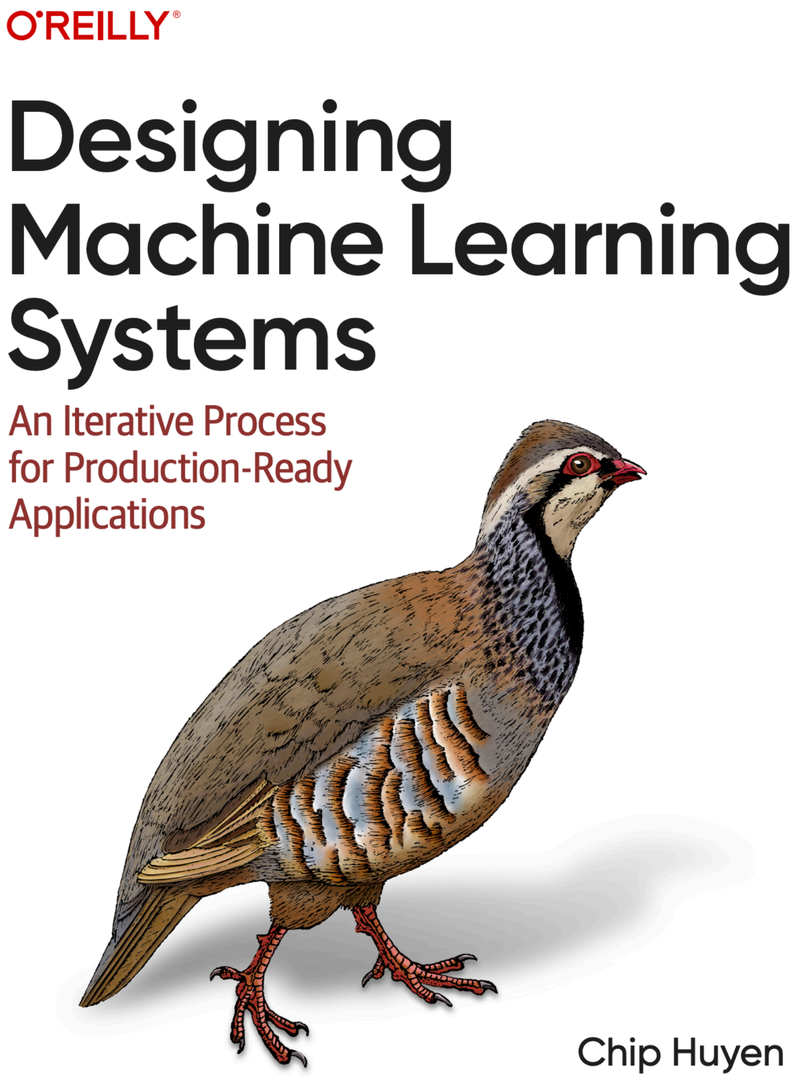
Checkout

- `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.

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

Case I

Using Machine Learning to Predict Value of Homes On Airbnb

by Robert Chang

Amazing view from a Airbnb Home in Imerovigli, Egeo, Greece

Using Machine Learning to Predict Value of Homes On Airbnb

- Airbnb used machine learning to predict a vital business metric: the value of homes on Airbnb.
- It walks you through the entire workflow: feature engineering, model selection, prototyping, and moving prototypes to production.
- It’s completed with lessons learned, tools used, and code snippets too.

Case II

Netflix Technology Blog

Mar 22, 2018 · 7 min read · Listen





Using Machine Learning to Improve Streaming Quality at Netflix

by Chaitanya Ekanadham

One of the common questions we get asked is: “Why do we need machine learning to improve streaming quality?” This is a really important question, especially given the recent hype around machine learning and AI which can lead to instances where we have a “solution in search of a problem.” In this blog post, we describe some of the technical challenges we face for video streaming at Netflix and how statistical models and machine learning techniques can help overcome these challenges.

Using Machine Learning to Improve Streaming Quality at Netflix

- As of 2018, Netflix streams to over 117M members worldwide, half of those living outside the US.
- This blog post describes some of their technical challenges and how they use machine learning to overcome these challenges, including:
  - predicting the network quality,
  - detect device anomaly,
  - and allocate resources for predictive caching.

Case III

150 successful machine learning models: 6 lessons learned at Booking.com

OCTOBER 7, 2019 ~ ADRIAN COLYER

150 successful machine learning models: 6 lessons learned at Booking.com
Bernadi et al., *KDD'19*

Here's a paper that will reward careful study for many organisations. We've previously looked at the deep penetration of machine learning models in the product stacks of leading companies, and also some of the pre-requisites for being successful with it. Today's paper choice is a wonderful summary of lessons learned integrating around 150 successful customer facing applications of machine learning at Booking.com. Oddly enough given the paper title, the six lessons are never explicitly listed or enumerated in the body of the paper, but they can be inferred from the division into sections. My interpretation of them is as follows:

150 Successful Machine Learning Models: 6 Lessons Learned at booking.com

- As of 2019, Booking.com has around 150 machine learning models in production.
  - Predicting users' travel preferences and how many people they travel with
  - Optimizing the background images and reviews to show for each user.
- Lessons Learned:
  - Machine-learned models deliver strong business value.
  - Model performance is not the same as business performance.
  - Be clear about the problem you're trying to solve.
  - Prediction serving latency matters.
  - Get early feedback on model quality.
  - Test the business impact of your models using randomized controlled trials.