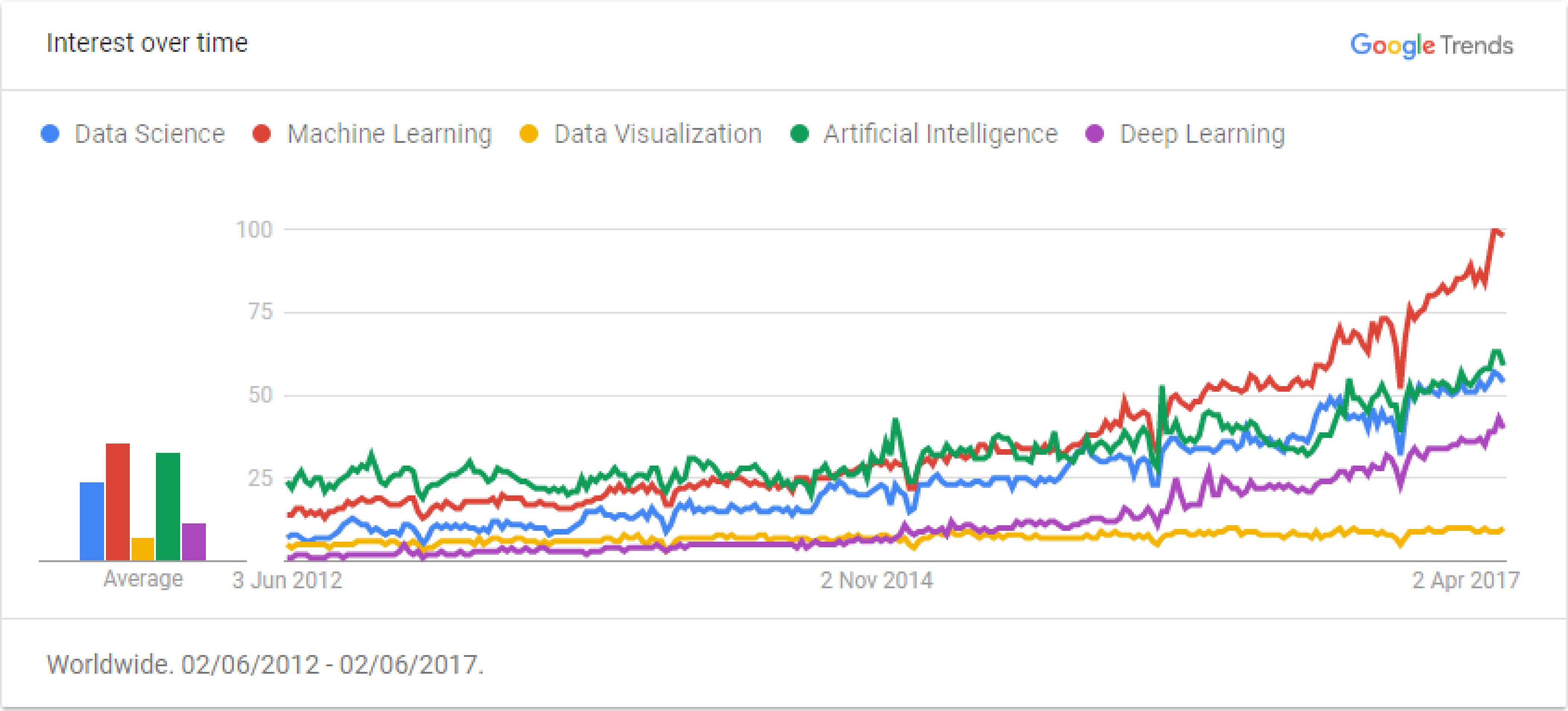
Buzzwords surrounding Data Science

Eiko Yoneki
eiko.yoneki@cl.cam.ac.uk
http://www.cl.cam.ac.uk/~ey204

Systems Research Group
University of Cambridge Dept. Computer Science and Technology
Computer Laboratory

Rise of Data Science



Outline

- Data Science Community
- Landscape of Data Science Research in Alan Turing Institute
  - What Topics are interesting to research?
- Becoming Data Scientist?
- Fill the Gap between Research and Practice

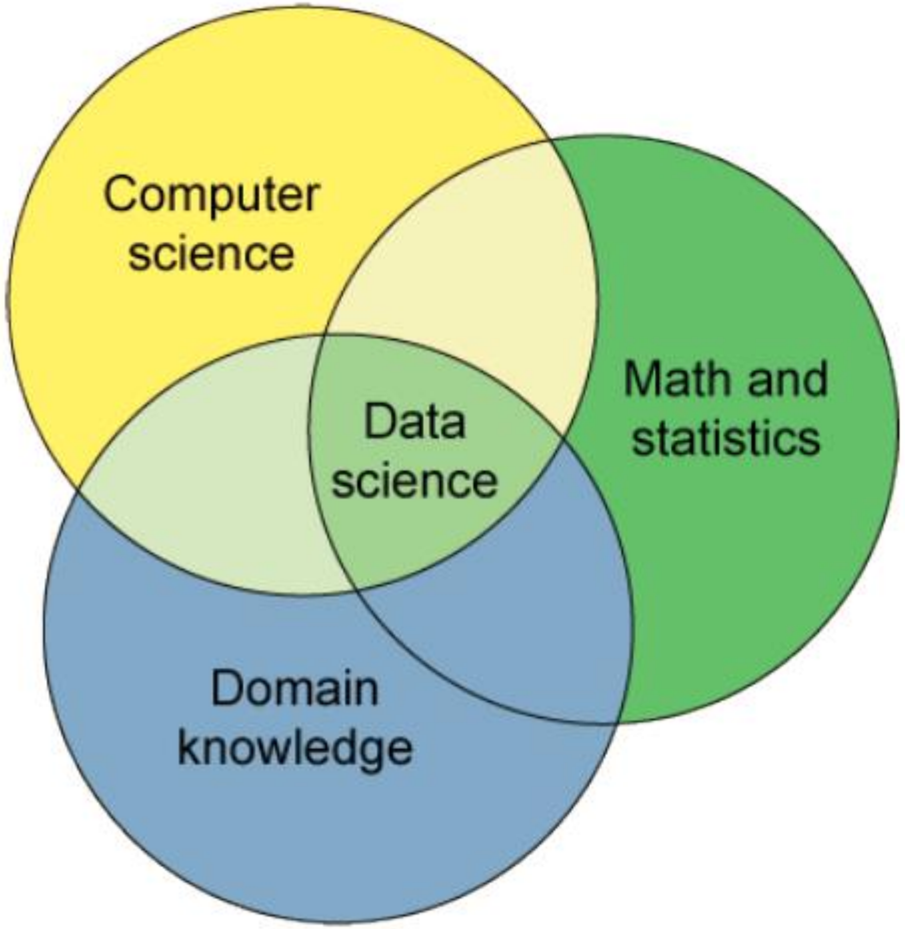
Data Science: Any new intellectual content?

- What does it mean to Computer Science?

- 1970's: EE + Math → Computer Science
- 2010's: CS + Stats + ?? → Data Science

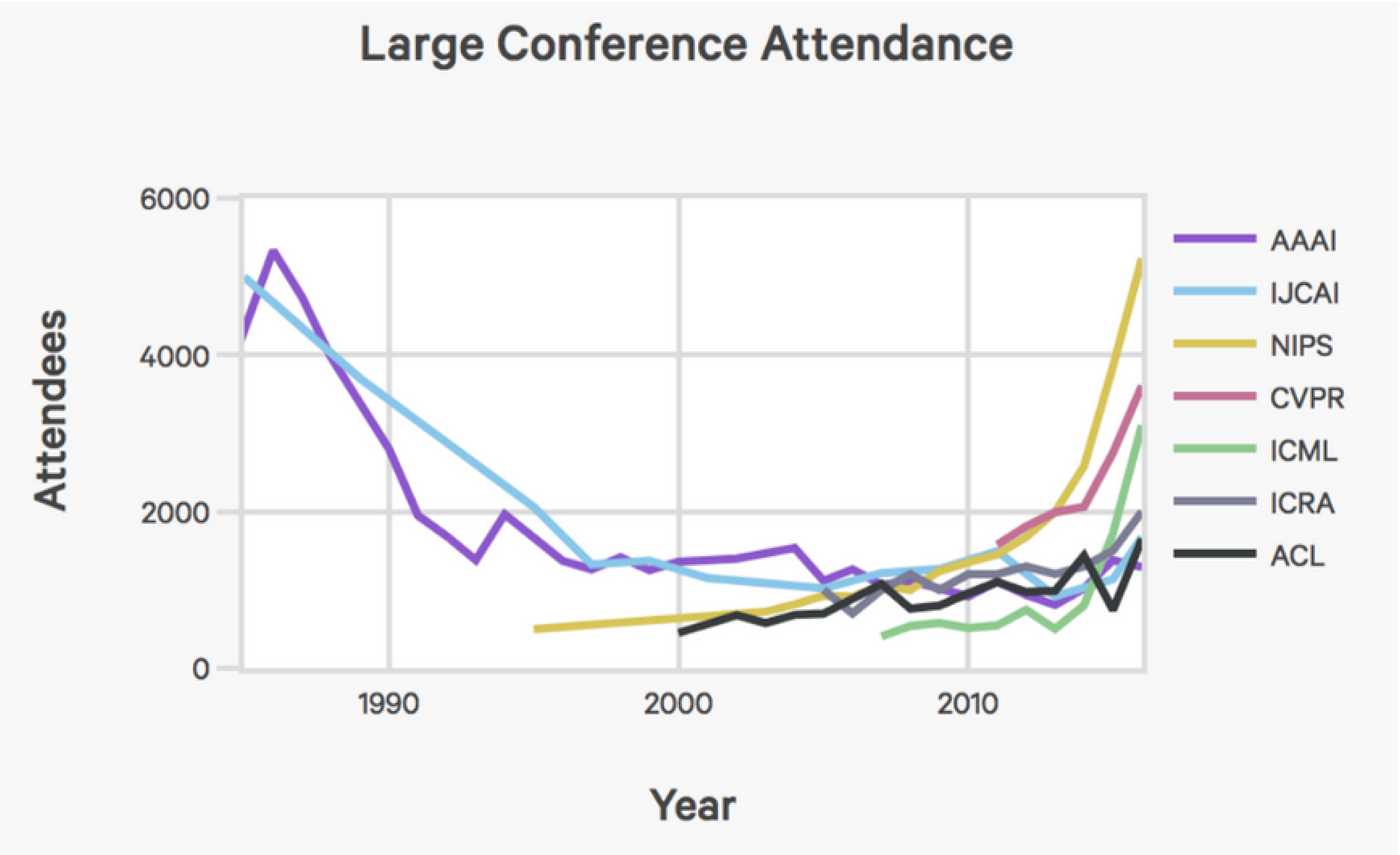
- Is something fundamental emerging here?

- Data Science is a very broad discipline
- Data Science PhD?
  - PhD normally with a narrow field with depth…



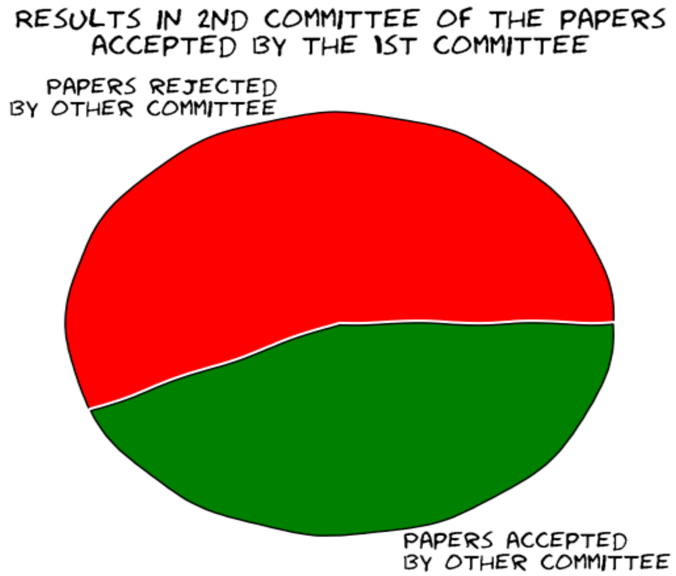
based on Drew Conway, NYU

Scale of Community Size in ML/AI

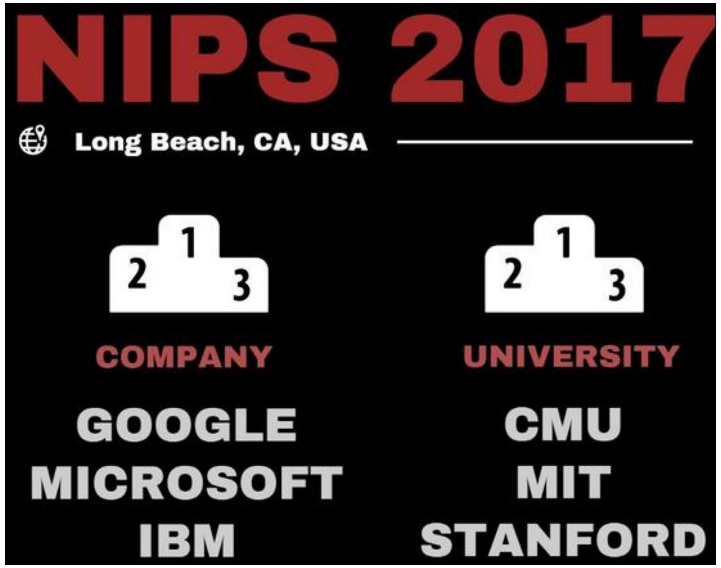


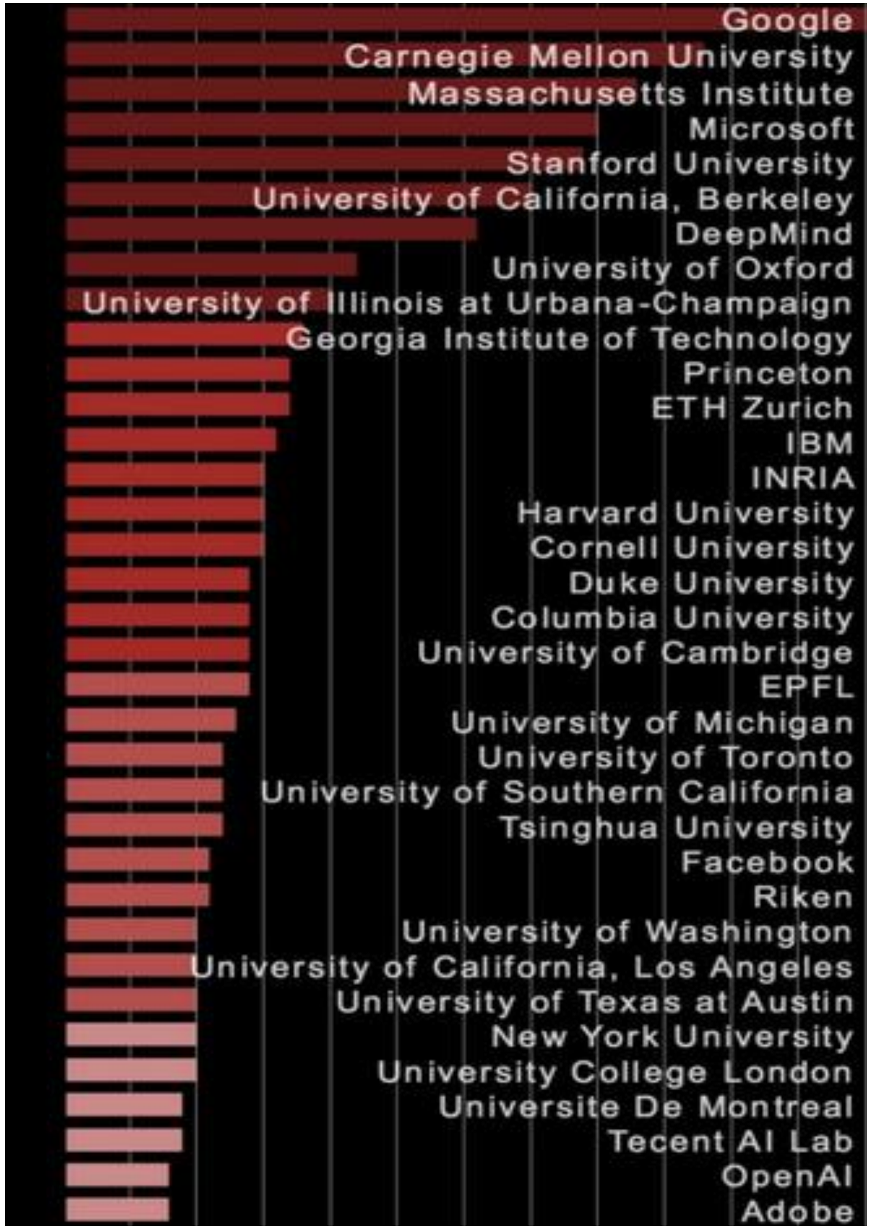
NIPS: 8000 Attendees in 2017

- Randomness of Paper acceptance?
- 2016: 2,406 submissions and 568 acceptance (24% acceptance rate)
- 2017: 3,240 submissions and 679 acceptance (21% acceptance rate)
- In 2014, Corinna Cortes and Neil Lawrence ran the NIPS experiment where 1/10th of papers submitted to NIPS went through the NIPS review process twice, and then the accept/reject decision was compared. http://blog.mrtz.org/2014/12/15/the-nips-experiment.html
- In particular, about 57% of the papers accepted by the first committee were rejected by the second one and vice versa. In other words, most papers at NIPS would be rejected if one reran the conference review process (with a 95% confidence interval of 40-75%).



NIPS: Publishing





SysML Conference spawn in 2018-2019

- SysML is a conference targeting research at the intersection of systems and machine learning
- Aims to elicit new connections amongst these fields, including identifying best practices and design principles for learning systems, as well as developing novel learning methods and theory tailored to practical machine learning workflows

Steering Committee

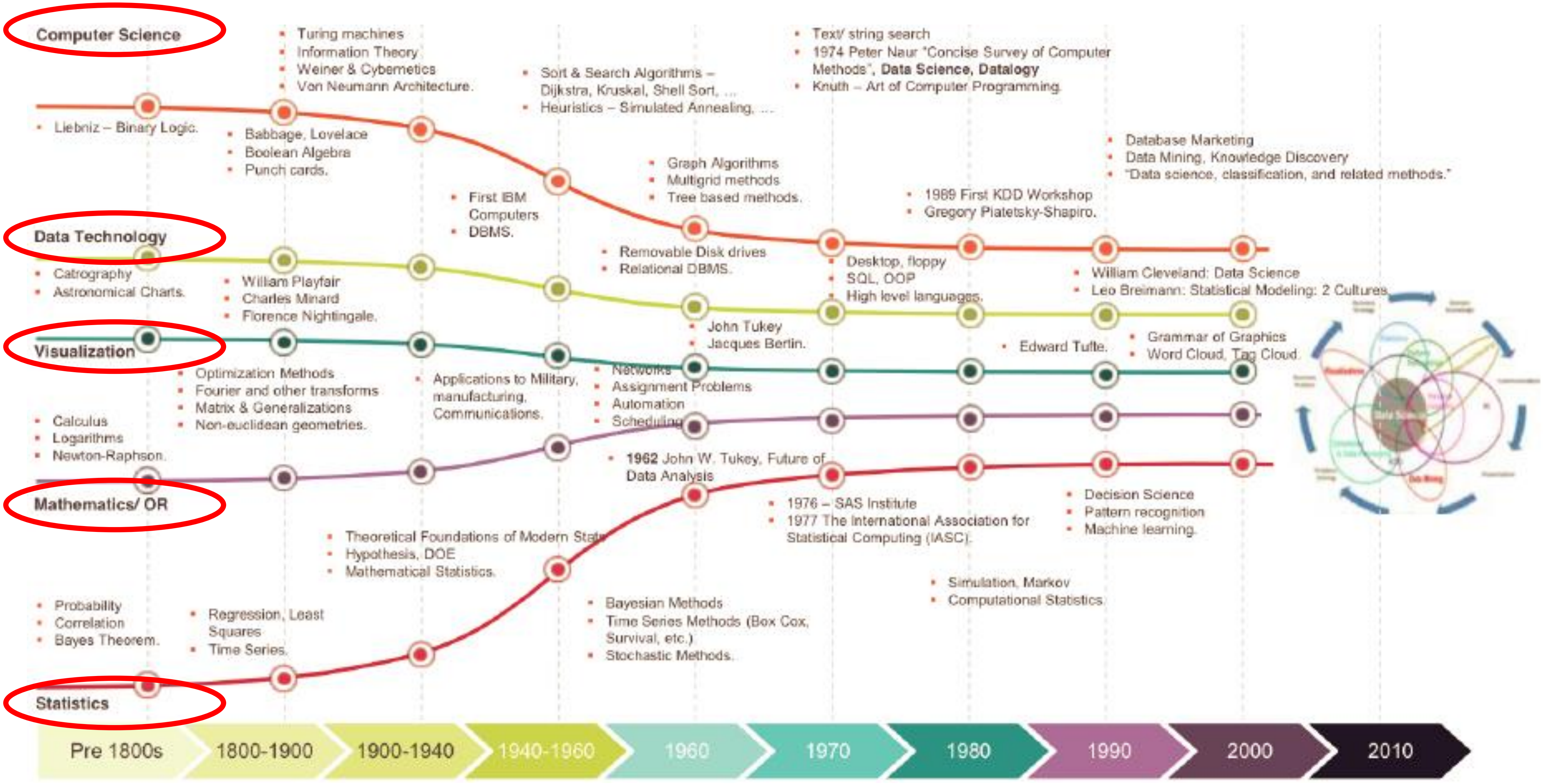
Jennifer Chayes
Bill Dally
Jeff Dean
Michael I. Jordan
Yann LeCun
Fei-Fei Li
Alex Smola
Dawn Song
Eric Xing

SysML 2019: Programming Models Session

Session IV: **Programming Models**

- RLgraph: Modular Computation Graphs for Deep Reinforcement Learning
  Michael Schaarschmidt (University of Cambridge), Sven Mika (rlcore), Kai Fricke (Helmut Schmidt University), Eiko Yoneki (University of Cambridge)
- TensorFlow Eager: A multi-stage, Python-embedded DSL for machine learning
  Akshay Agrawal (Google Brain), Akshay Naresh Modi (Google Brain), Alexandre Passos (Google Brain), Allen Lavoie (Google Brain), Ashish Agarwal (Google Brain), Asim Shankar (Google Brain), Igor Ganichev (Google Brain), Josh Levenberg (Google Brain), Mingsheng Hong (Google Brain), Rajat Monga (Google Brain), Shanqing Cai (Google Brain)
- AutoGraph: Imperative-style Coding with Graph-based Performance
  Dan Moldovan (Google Inc.), James Decker (Purdue University), Fei Wang (Purdue University), Andrew Johnson (Google Inc.), Brian Lee (Google Inc.), Zack Nado (Google Inc.), D Sculley (Google), Tiark Rompf (Purdue University), Alexander B Wiltschko (Google Inc.)
- TensorFlow.js: Machine Learning for the Web and Beyond
  Daniel Smilkov (Google), Nikhil Thorat (Google), Yannick Assogba (Google), Charles Nicholson (Verily), Nick Kreeger (Google), Ping Yu (Google), Shanqing Cai (Google), Eric Nielsen (Google), David Soegel (Google), Stan Bileschi (Google), Michael Terry (Google), Ann Yuan (Google), Kangyi Zhang (Google), Sandeep Gupta (Google), Sarah Sirajuddin (Google), D Sculley (Google), Rajat Monga (Google), Greg Corrado (Google), Fernanda Viegas (Google), Martin M Wattenberg (Google)

History/Trajectory Data Science



Alan Turing Institute (ATI)

- Established in 2015 in London as a National Institute for Data Science
- >£20M Capital Investment from Government
- Originally 5 Universities formed core body (UCL, Warwick, Edinburg, Oxford and Cambridge) and now expanded to 13 and more universities
- Goal: Data Science and after 2018 changed to Artificial Intelligence

https://www.turing.ac.uk

Driving data futures: Technology and government – the good, the bad, and the ugly →
Thursday 02 May 2019
Time: 17:15 - 19:00

Lina Dencik | Omar A Guerrero

Turing Lecture: Learning how to learn efficiently →
Tuesday 30 Apr 2019
Time: 18:00 - 20:30

Nando de Freitas

Mathematics of data →
Wednesday 29 May 2019 - Friday 31 May 2019
Time: 10:00 - 17:00

ATI: Research Programmes

- Translating output into practice

Artificial intelligence (AI) →
Advancing world-class research into artificial intelligence, its applications and its implications for society, building on our academic network's wealth of expertise.

Data science at scale →
Building upon advances in high-performance computer architectures, through algorithm-architecture co-design, with applications including health and life science.

Data science for science →
Ensuring that research across science and the humanities can make effective use of state of the art methods in artificial intelligence and data science.

Health and medical sciences →
Accelerating the scientific understanding of human disease and improving human health through data-driven innovation in AI and statistical science.

Research Engineering →
Connecting research to applications, helping create usable and sustainable tools, practices and systems.

Data-centric engineering →
Bringing together world-leading academic institutions and major industrial partners from across the engineering sector, to address new challenges in data-centric engineering.

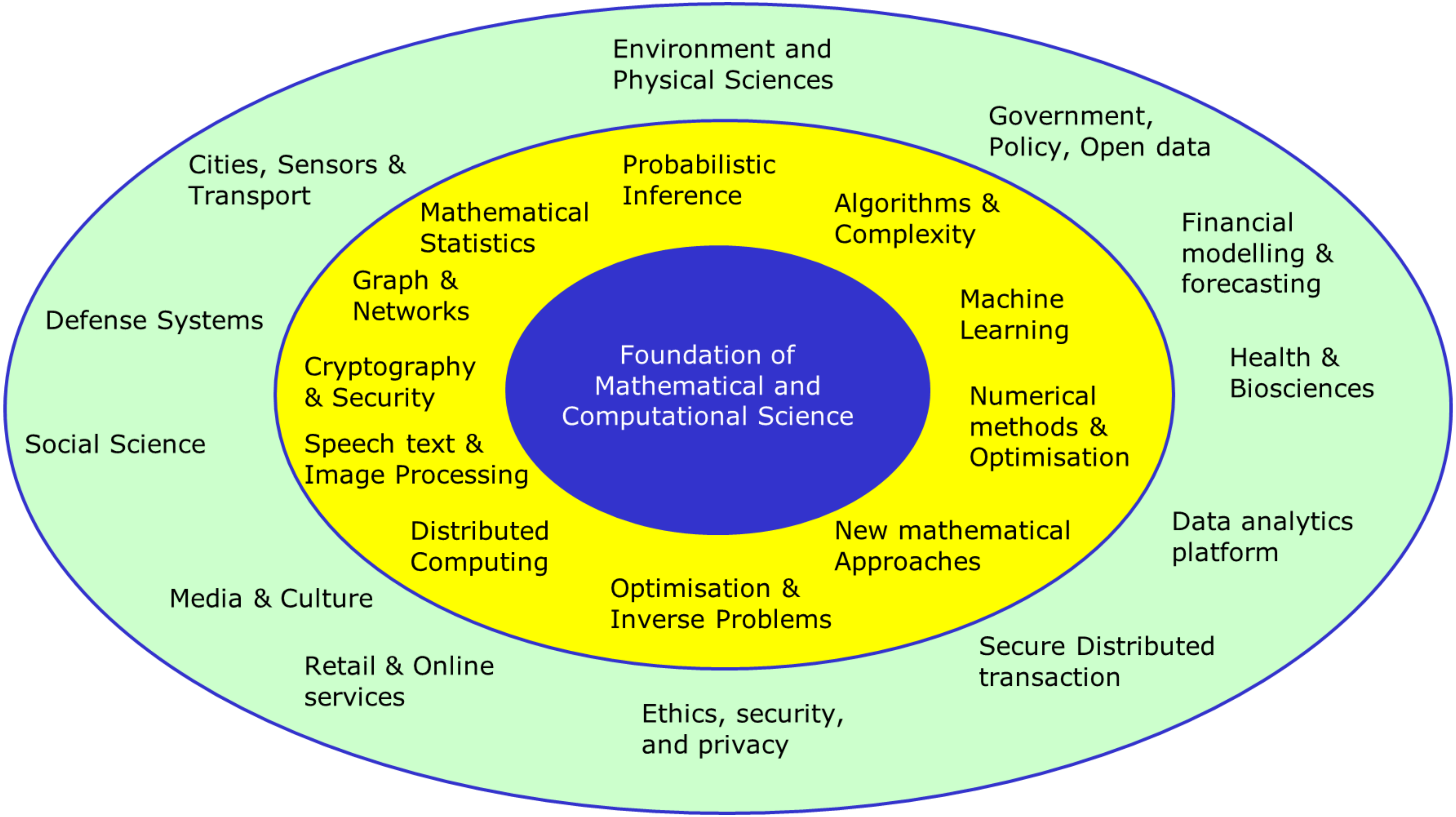
Defence and security →
Collaborating with the defence and security community to deliver an ambitious programme of data science research, to deliver impact in real world scenarios.

Finance and economics →
Develop cutting-edge methods to foster financial innovation and deepen our understanding of the economy, to benefit society at large

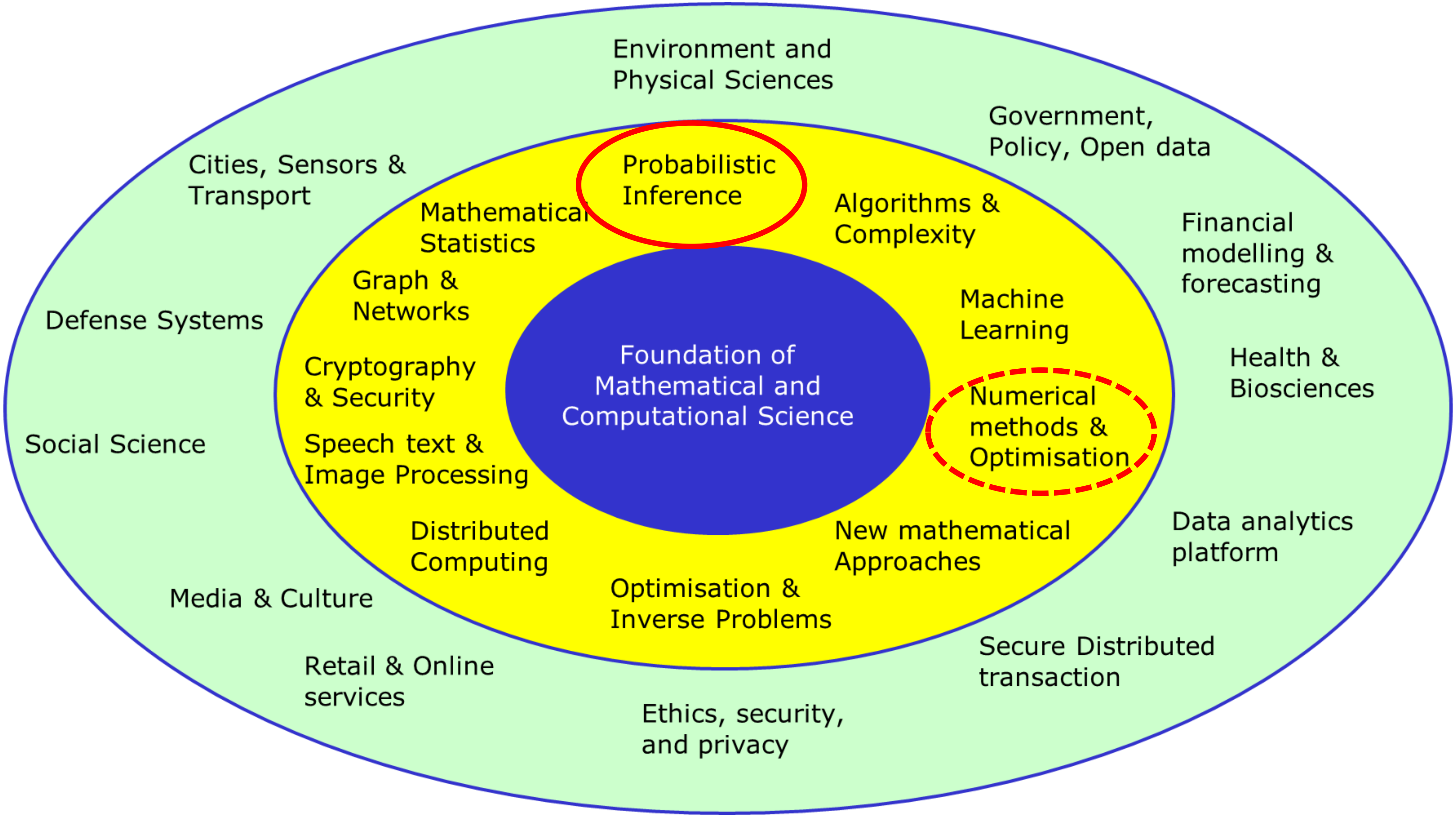
Urban analytics →
Developing data science and AI focused on the process, structure, interactions and evolution of agents, technology and infrastructure within and between cities.

Public policy →
Working with policy makers on data-driven public services and innovation to solve policy problems, and developing ethical foundations for data science and AI policy-making.

Landscape of Data Science Research in ATI



Landscape of Data Science Research in ATI



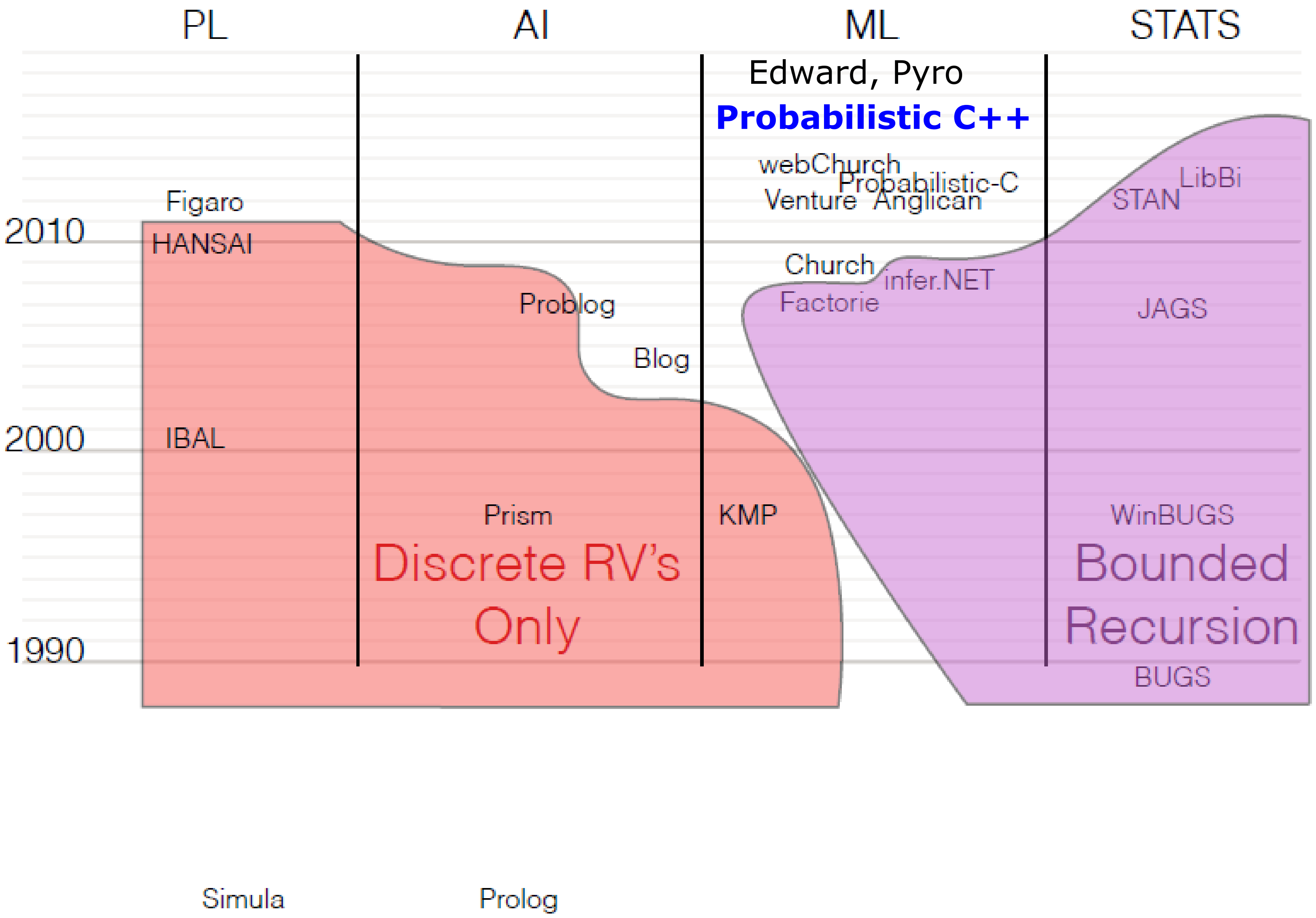
Probabilistic Model

- Probabilistic models incorporate random variables and probability distributions into the model
  - Deterministic model gives a single possible outcome
  - Probabilistic model gives a probability distribution
- Used for various probabilistic logic inference (e.g. MCMC-based inference, Bayesian inference…)

Python based PP:

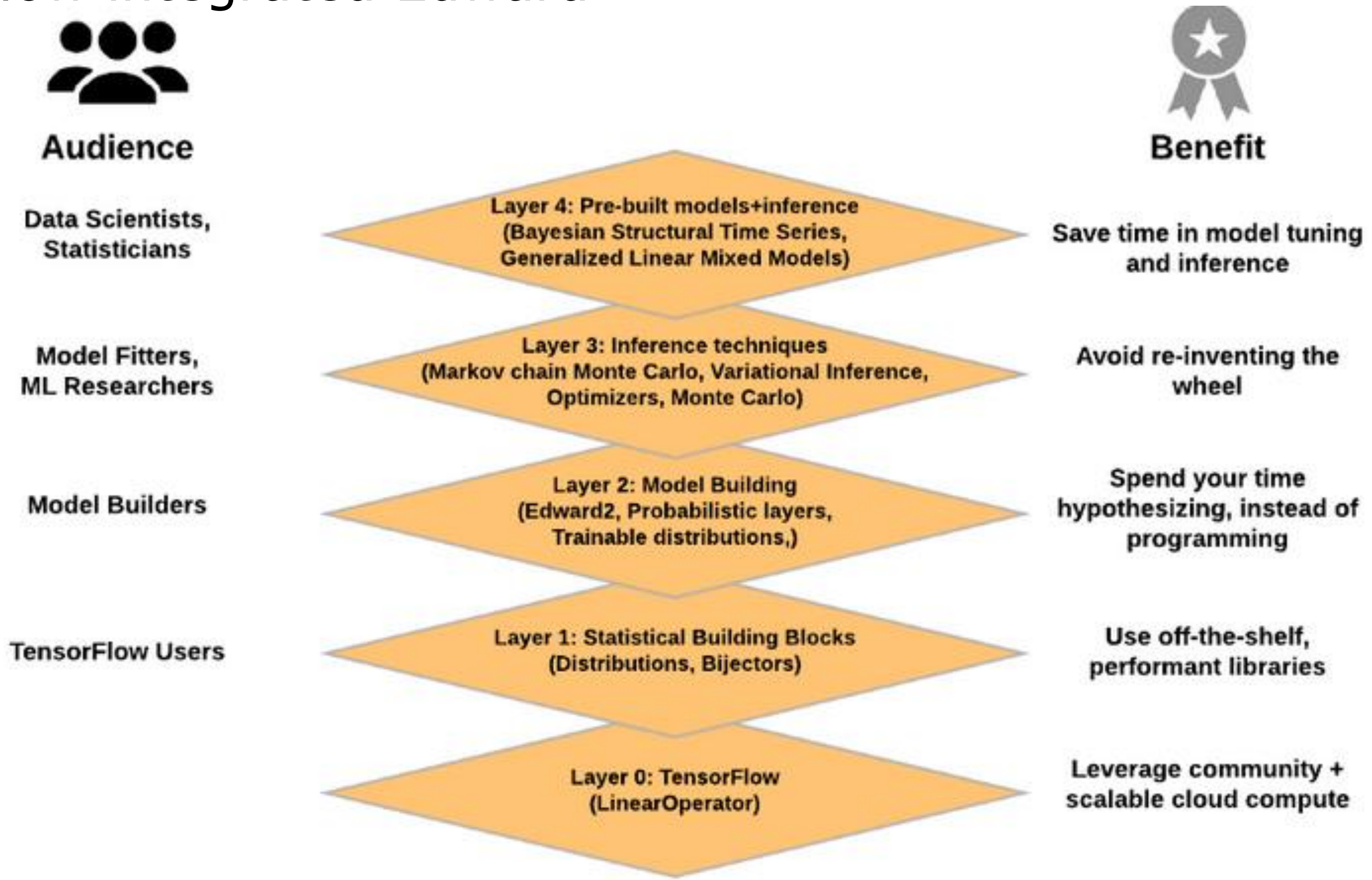
- Pyro: https://pyro.ai/examples
- Edward: http://edwardlib.org

Probabilistic Programming

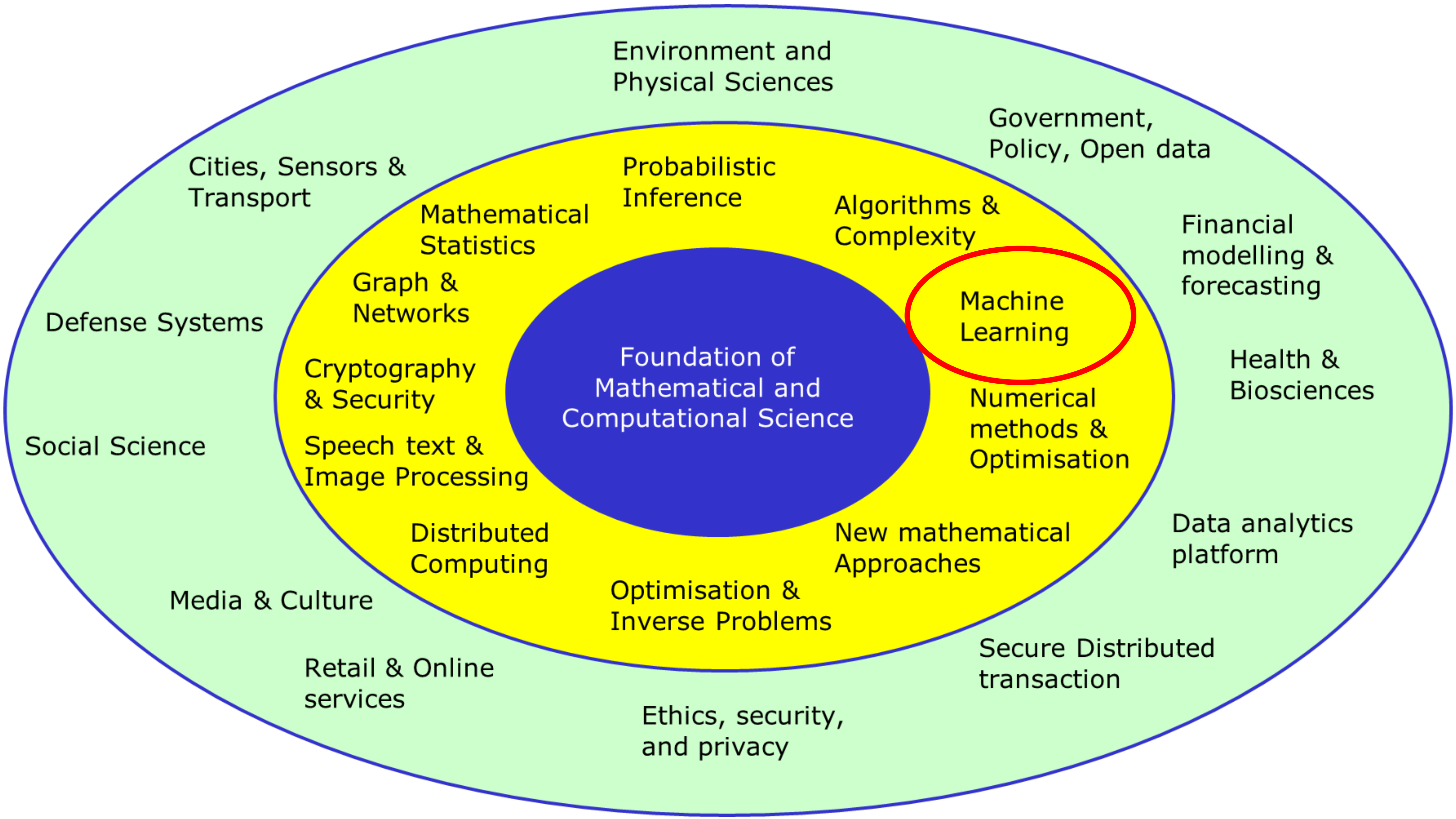


TensorFlow Probability

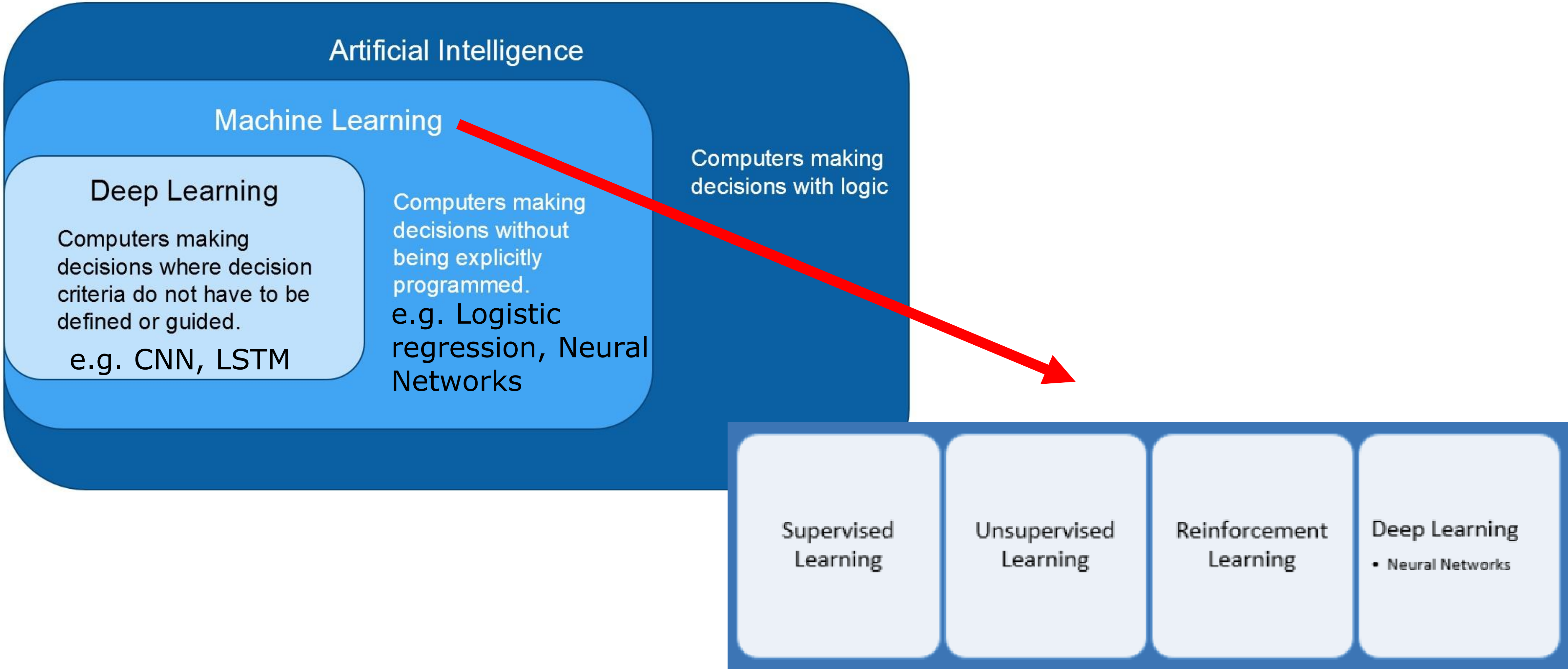
TensorFlow integrated Edward



Landscape of Data Science Research in ATI



Deep Learning, Machine Learning, and AI…



Machine Learning Timeline

- Modern Machine Learning: see Wikipedia-Timeline of machine learning

2009	Achievement	ImageNet	ImageNet is created. ImageNet is a large visual database envisioned by Fei-Fei Li from Stanford University, who realized that the best machine learning algorithms wouldn't work well if the data didn't reflect the real world.[40] For many, ImageNet was the catalyst for the AI boom[41] of the 21st century.
2010		Kaggle Competition	Kaggle, a website that serves as a platform for machine learning competitions, is launched.[42]
2011	Achievement	Beating Humans in Jeopardy	Using a combination of machine learning, natural language processing and information retrieval techniques, IBM's Watson beats two human champions in a Jeopardy! competition.[43]
2012	Achievement	Recognizing Cats on YouTube	The Google Brain team, led by Andrew Ng and Jeff Dean, create a neural network that learns to recognize cats by watching unlabeled images taken from frames of YouTube videos.[44][45]
2014		Leap in Face Recognition	Facebook researchers publish their work on DeepFace, a system that uses neural networks that identifies faces with 97.35% accuracy. The results are an improvement of more than 27% over previous systems and rivals human performance.[46]
2014		Sibyl	Researchers from Google detail their work on Sibyl,[47] a proprietary platform for massively parallel machine learning used internally by Google to make predictions about user behavior and provide recommendations.[48]
2016	Achievement	Beating Humans in Go	Google's AlphaGo program becomes the first Computer Go program to beat an unhandicapped professional human player[49] using a combination of machine learning and tree search techniques.[50] Later improved as AlphaGo Zero and then in 2017 generalized to Chess and more two-player games with AlphaZero.

Four Great Pictures Illustrating ML Concepts

- Neural Networks: The Backpropagation algorithm
- Cheat Sheet on Probability
- 24 Neural Network Adjustments
- Matrix Multiplication in NN

ARCHITECTURE

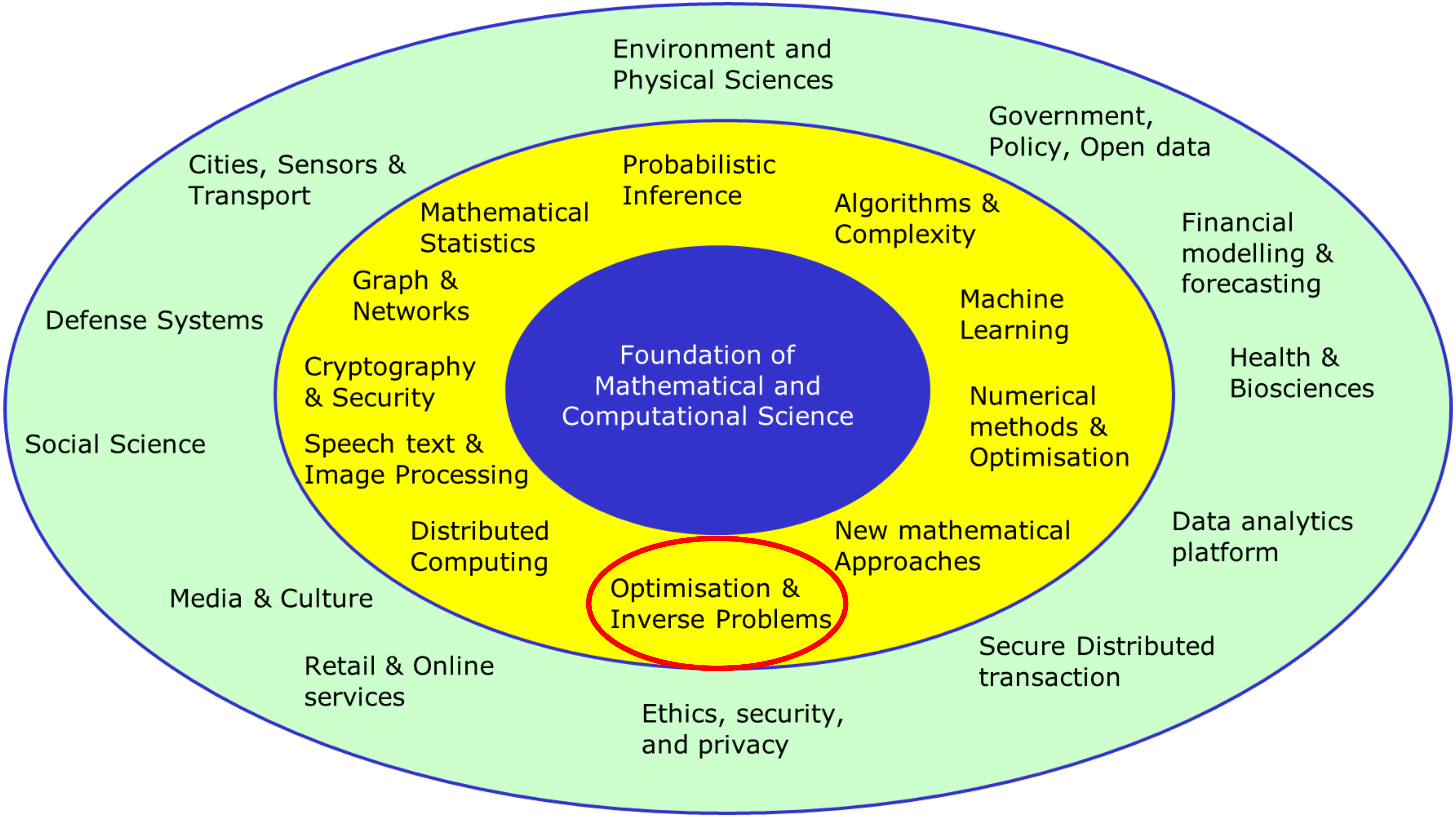
- Variables type
- Variable scaling
- Cost function
- Neural Network type:
  - RBM,FFN,CNN,RNN...
- Number of layers
- Number of hidden Layers
- Number of nodes
- Type of layers:
  - LSTM, Dense, Highway
  - Convolutional, Pooling...
- Type of weight initialization
- Type of activation function
  - Linear, sigmoid, relu...
- Dropout rate (or not)
- Threshold

HYPERPARAMETER TUNING

- Type of optimizer
- Learning rate (fixed or not)
- Regularization rate (or not)
- Regularization type: L1, L2, ElasticNet
- Type of search for local minima:
  - Gradient descent, simulated
  - annealing, evolutionary...
- Batch size
- Nesterov momentum (or not)
- Decay rate (or not)
- Momentum (fixed or not)
- Type of fitness measurement:
  - MSE, accuracy, MAE, cross-entropy,
  - precision, recall
- Epochs
- Stop criteria

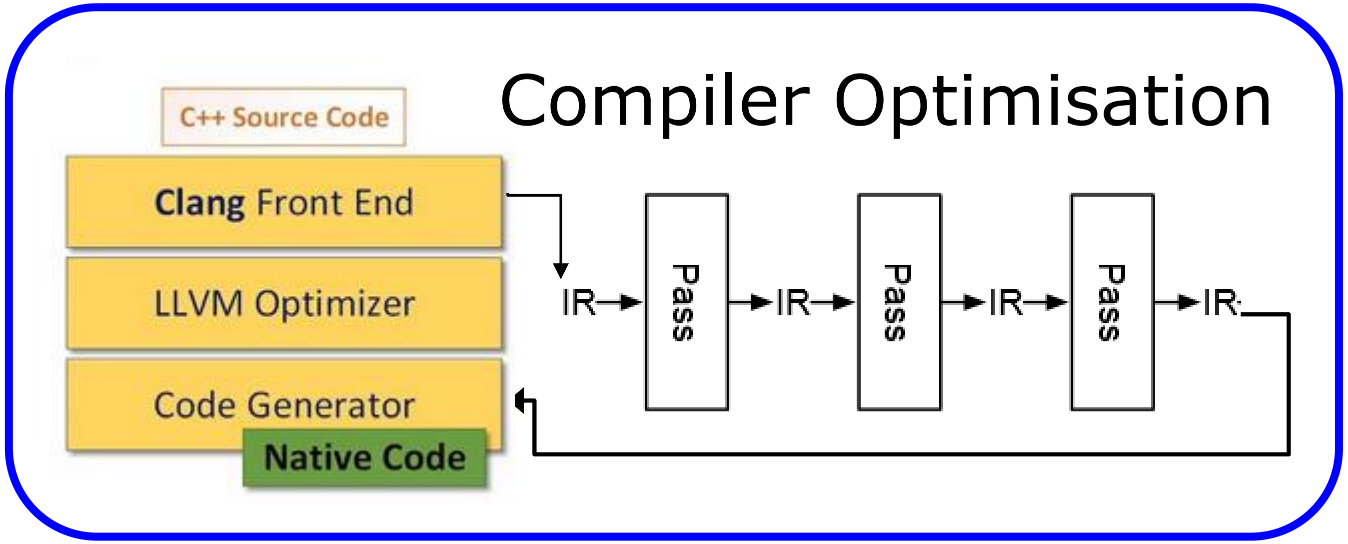
See https://www.datasciencecentral.com/profiles/blogs/four-great-pictures
illustrating-machine-learning-concepts

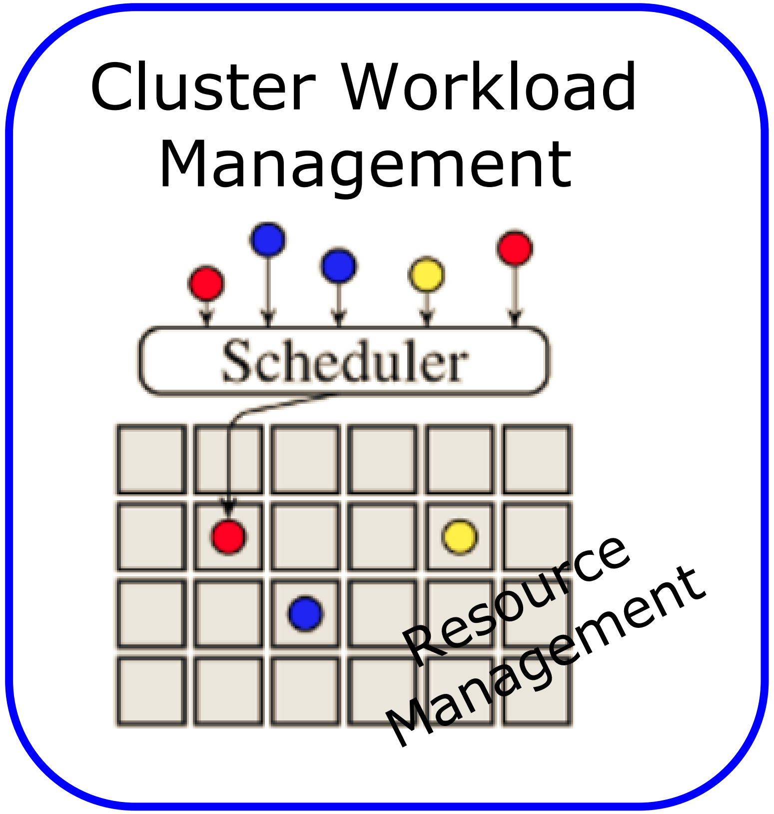
Landscape of Data Science Research in ATI



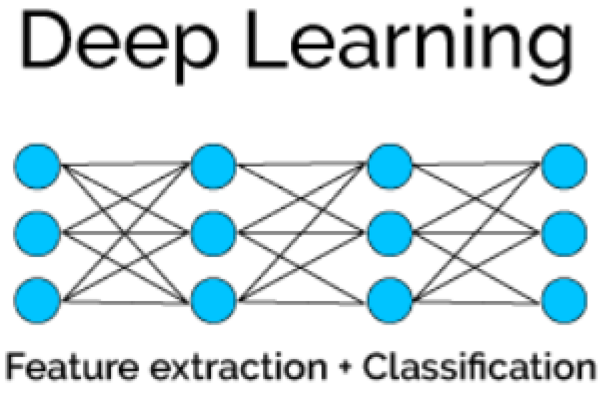
Tuning Computer Systems is Complex

- Complex configuration parameter space / increasing # of parameters
- Configurations need tuning to optimise resource utilisation
- Hand-crafted solutions impractical, often left static or configured through extensive offline analysis



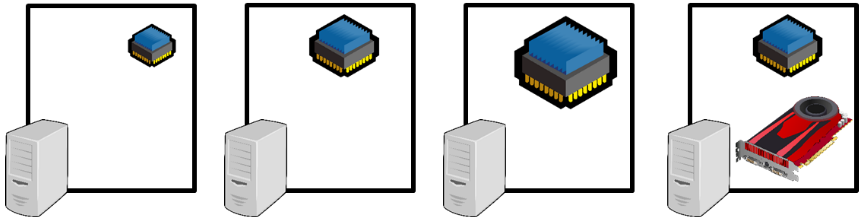


Complex and High Dimension Parameter Space



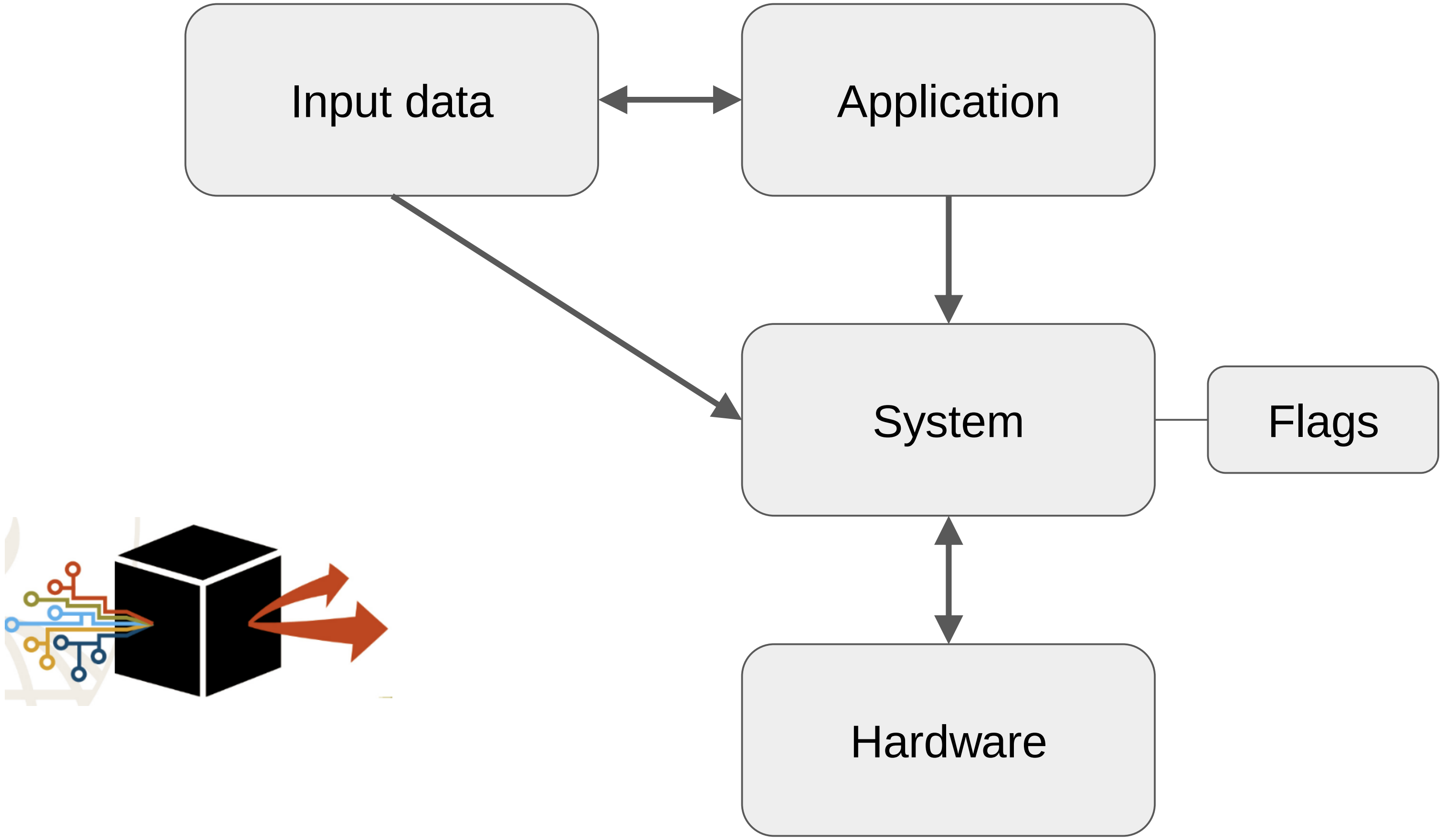
Hyper-Parameters:
- Learning-rate
- Number of Dense Layers
- Number of Dense Nodes
- Activation Function

Device Allocation
for Distributed Training



UBER

Auto-tuning systems



- Properties:
  - Many dimensions
  - Expensive objective function
  - Understanding of the underlying behaviour

Auto-tuning: Large Parameter Space

- Grid search $\theta \in [1, 2, 3, \ldots]$
- Evolutionary approaches (e.g. PetaBricks)
- Hill-climbing (e.g. OpenTuner)
- Bayesian optimization (e.g. SPEARMINT)

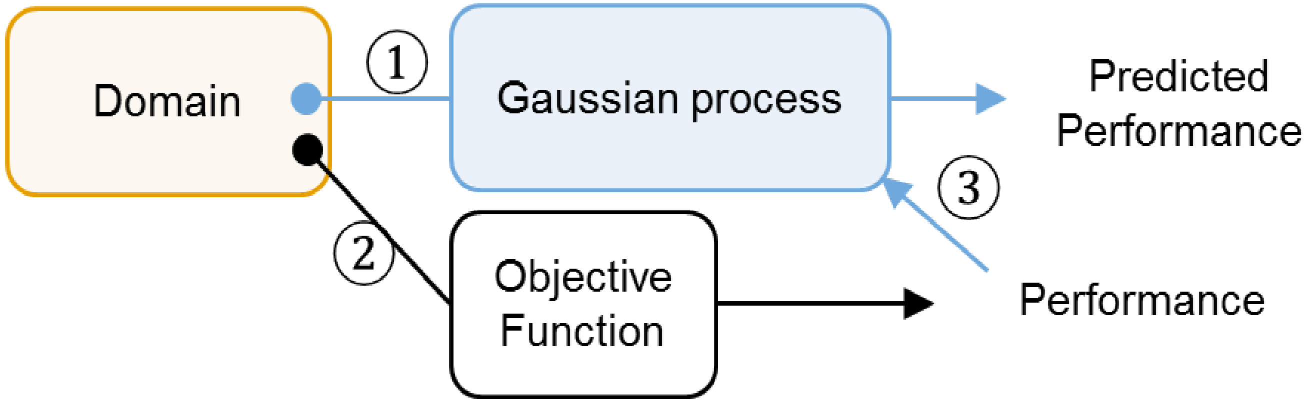
Require 1000s of evaluations of cost function!

Fails in high dimensions!

Fewer samples

Bayesian optimisation

Iteratively build a probabilistic model of objective function



① Find promising point (parameter values with high performance value in the model)

② Evaluate the objective function at that point

③ Update the model to reflect this new measurement

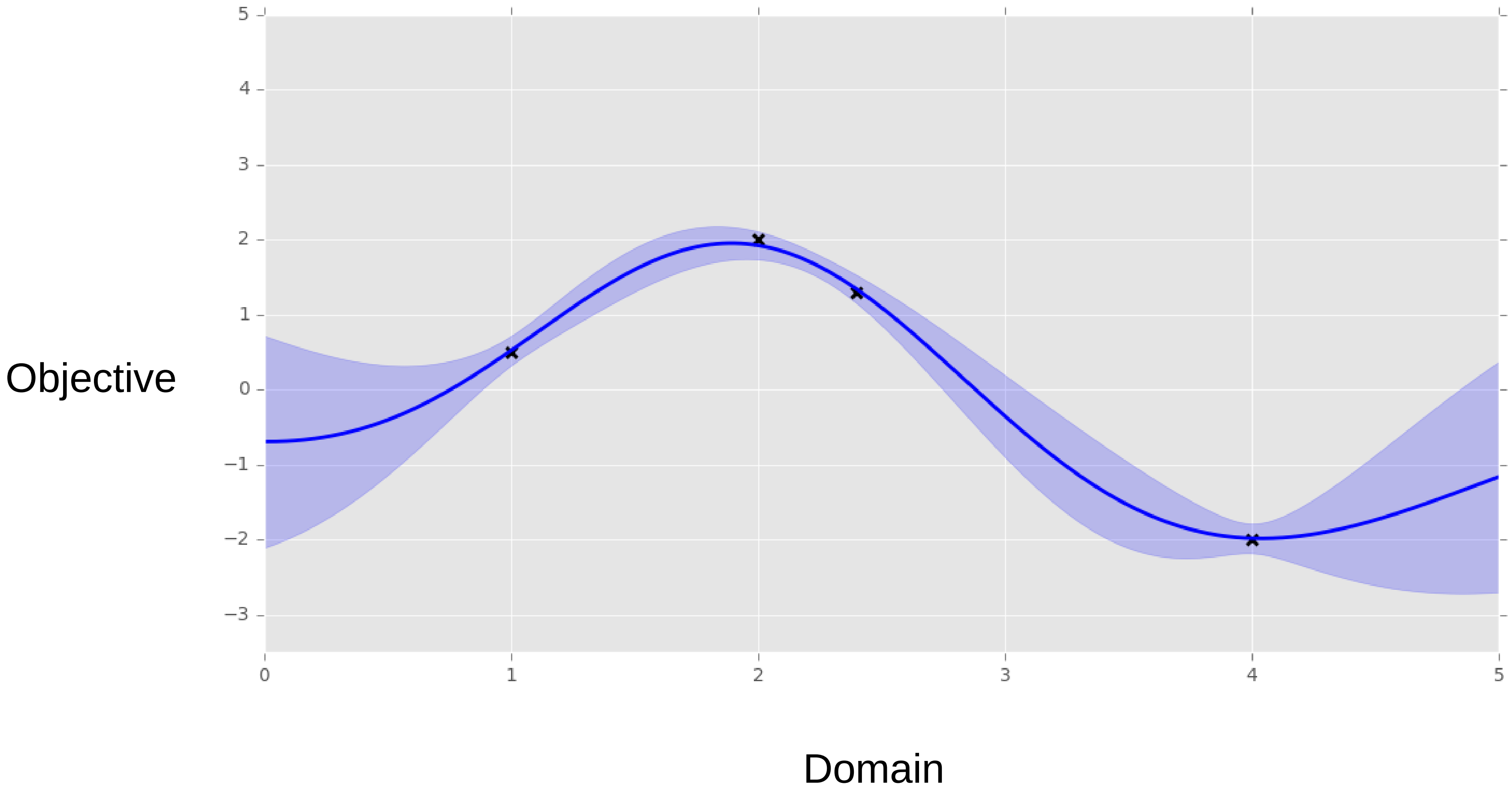
Pros:

- ✓ Data efficient: converges in few iterations
- ✓ Able to deal with noisy observations

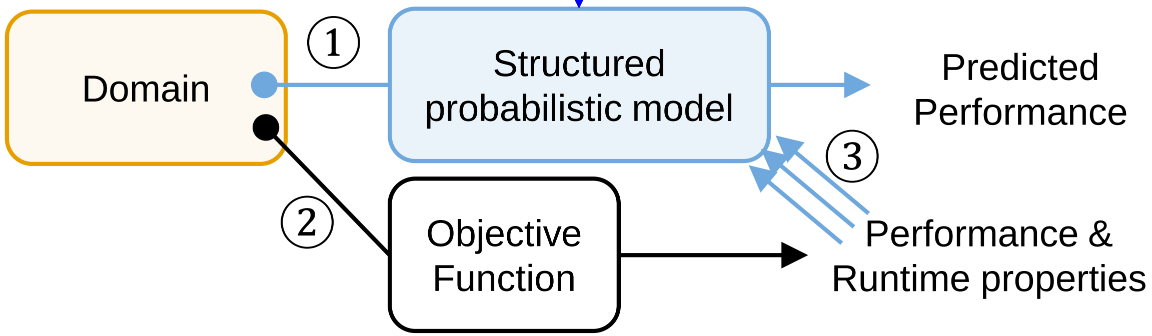
Cons:

- ✗ In many dimensions, model does not converge to the objective function

Bayesian optimisation



Structured Bayesian Optimisation (SBO)



Probabilistic Model written in Probabilistic C++

```
struct CassandraModel : public DAGModel<CassandraModel> {

  void model(int ygs, int sr, int mtt){
    // Calculate the size of the heap regions
    double es = ygs * sr / (sr + 2.0);// Eden space's size
    double ss = ygs / (sr + 2.0);      // Survivor space's size

    // Define the dataflow between semi-parametric models
    double rate =     output("rate", rate_model, es);
    double duration = output("duration", duration_model,
                             es, ss, mtt);
    double latency =  output("latency", latency_model,
                             rate, duration, es, ss, mtt);
  }

  ProbEngine<GCRateModel> rate_model;
  ProbEngine<GCDurationModel> duration_model;
  ProbEngine<LatencyModel> latency_model;
};
```

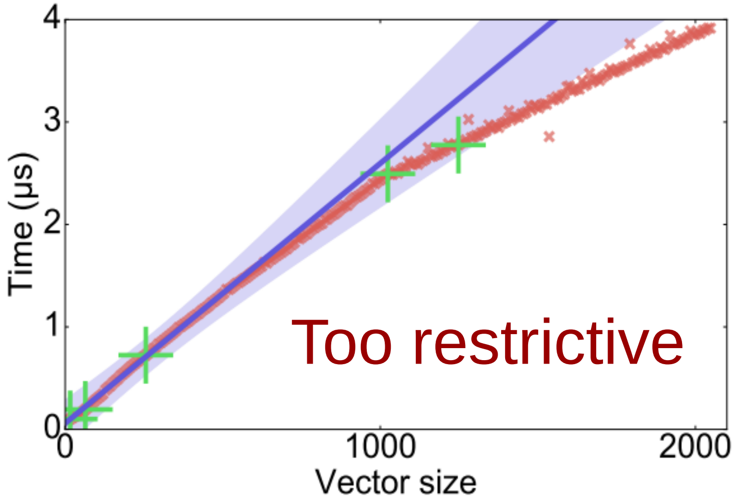
Developer-specified, model of performance from observed performance + arbitrary runtime characteristics

✓ Better convergence
✓ Use all measurements

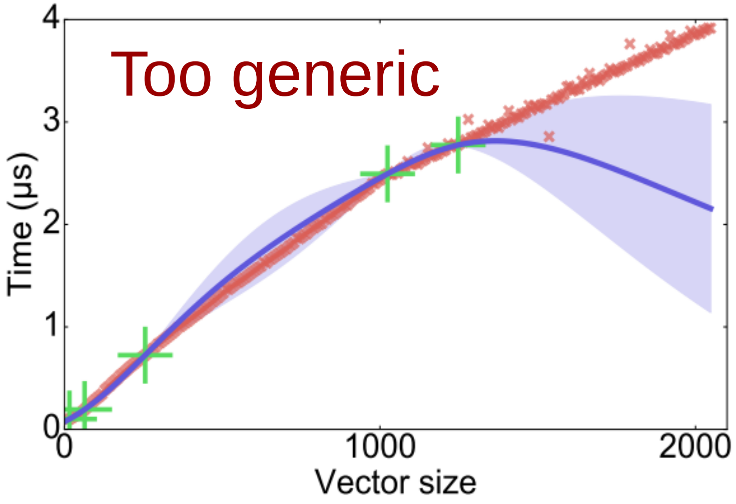
- **BOAT:** a framework to build **B**esp**O**ke **A**uto-**T**uners
- It includes a probabilistic library to express these models

Semi-parametric Model

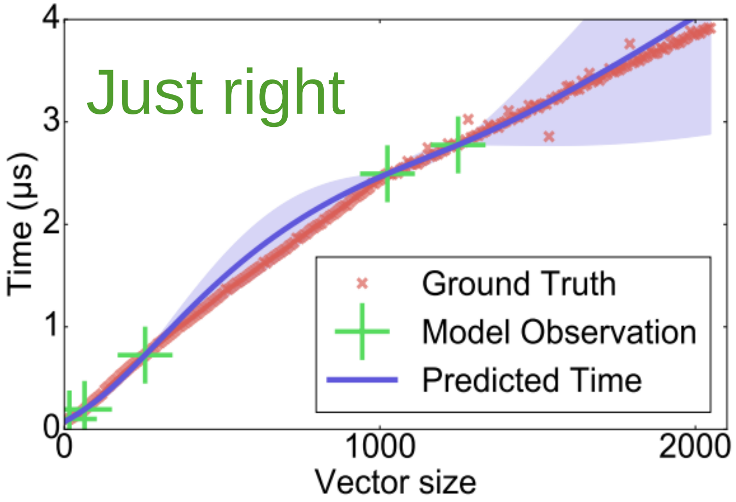
- Easy to use and well suited to SBO
  - Understand general trend of Objective function
  - High precision in region of Optimum for finding highest performance



(a) Parametric (Linear regression)



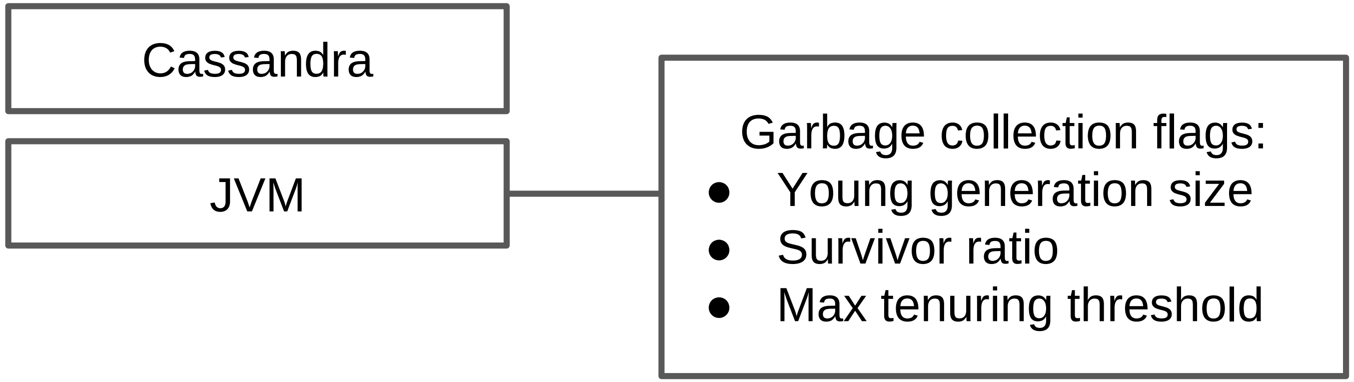
(b) Non-parametric (Gaussian process)



(c) Semi-parametric (Combination)

Example:

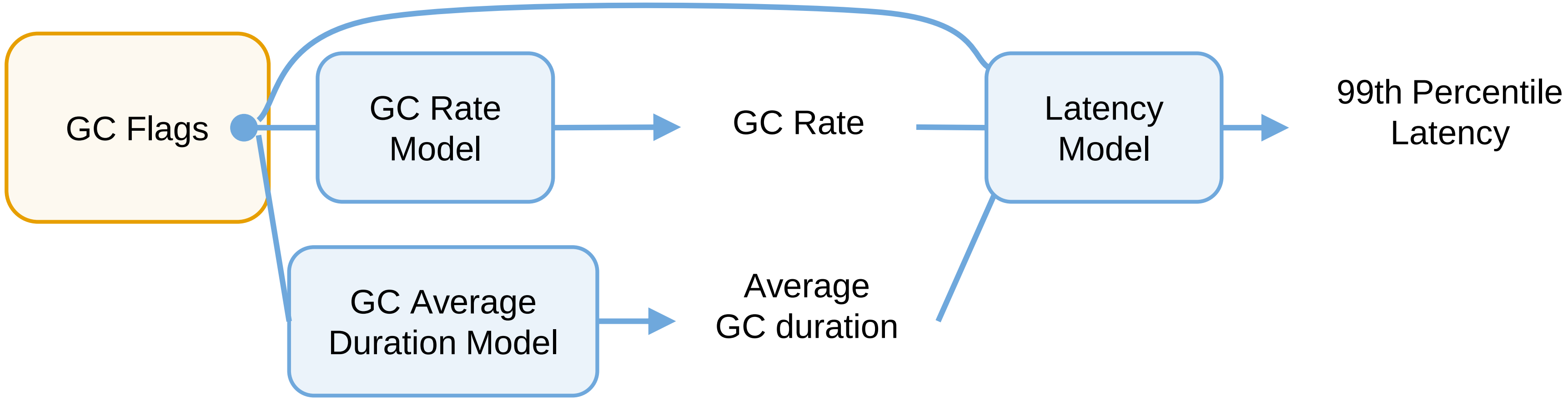
- Cassandra's garbage collection



- Minimise 99th percentile latency of Cassandra

Define DAG Model

- Define a directed acyclic graph (DAG) of models



Tune JVM parameters of a database (Cassandra) to minimise latency

Computer Systems Optimisation Models

- ***Short-term dynamic control:*** major system components are under dynamic load, such as resource allocation and stream processing, where the future load is not statistically dependent on the current load.
  BaysOpt is sufficient to optimise distinct workloads. For dynamic workload, Reinforcement Learning would perform better.

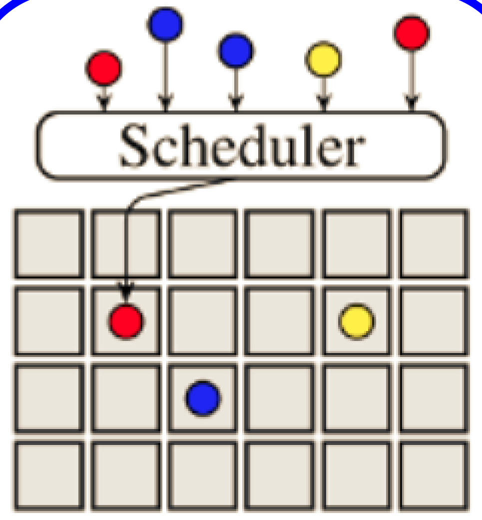
- ***Combinatorial optimisation:*** a set of options to be selected from a larger set under potential rules of combination. There is no straightforward similarity between different combinations. Many problems in device assignment, indexing, compiler optimisation fall in this category.
  BaysOpt cannot be easily applied. Either learning online if the task is cheap via random sampling, or via RL + pre-training if the task is expensive, or massively parallel online training if the resources are available.

Many systems problems are combinatorial in nature

Problem: Controlling dynamic behaviour

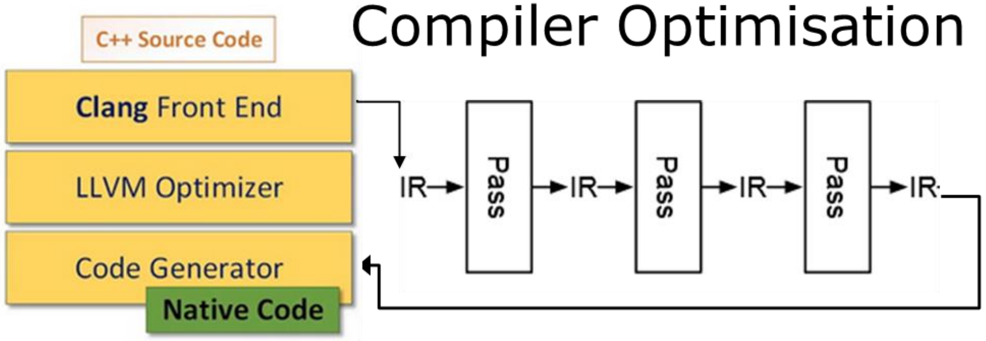
Assume workload dynamic, e.g. seasonality, load spikes, shared resources, failures..

- Algorithm: workload → behavior **distribution**
- Involves approximations to NP-complete problems, e.g. bin packing, sub-graph isomorphism, ..



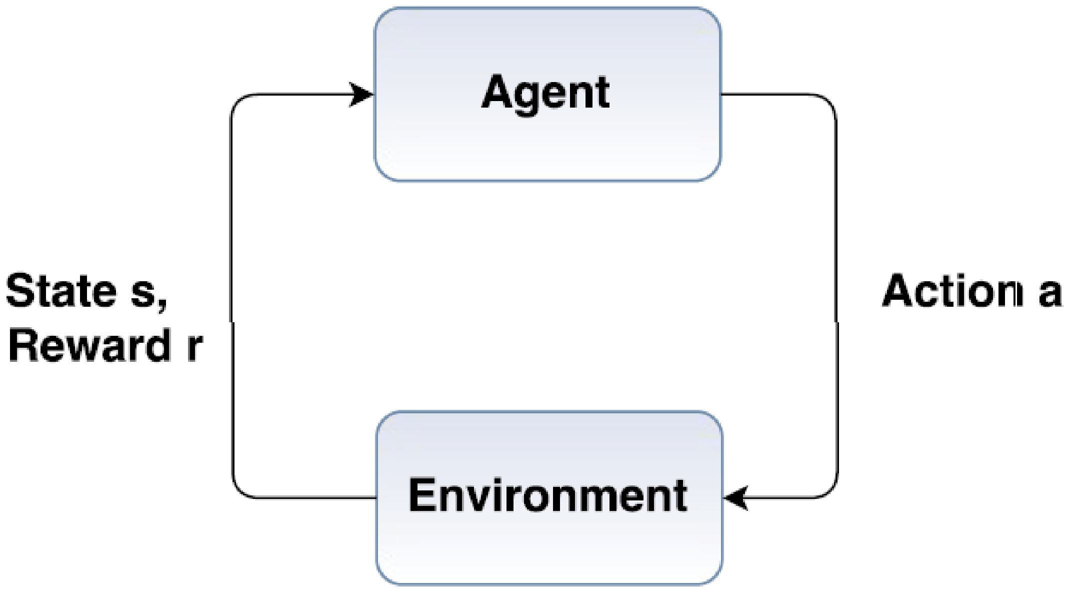
Source: firmament.io

Compiler Optimisation



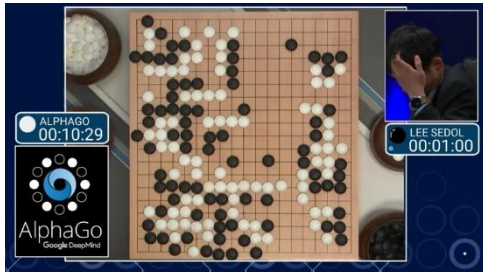
Reinforcement Learning

- Agent interacts with Dynamic environment
- Goal: Maximise expectations over rewards over agent's lifetime
- Notion of Planning/Control, not single static configuration



What makes RL different from other ML paradigms?

- There is no supervisor, only a reward signal
- Feedback is delayed, not instantaneous
- Time really matters (sequential)
- Agent's actions affect the subsequent data it receives



The most similar way to human brain's behaviour…

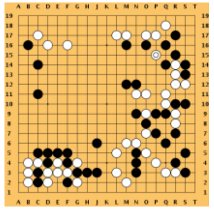
A brief history of Deep RL software

1. **Gen (2014-16):** Loose research scripts (e.g. DQN), high expertise required, only specific simulators

2. **Gen (2016-17):** OpenAI gym gives unified task interface, reference implementations (e.g. OpenAI baselines)

3. **Gen (2017-18):** Generic declarative APIs, distributed abstractions (Ray RLlib), some standard *flavours* emerge

Problems: Tightly coupled execution/logic, testing, reuse,..



RL Workloads

- Unlike supervised learning, not a single dominant execution pattern
- Distributed workloads: Hierarchies of sync/async data exchange
- Algorithms highly sensitive to hyper-parameters
- From large scale parallel training (e.g. AlphaGo) to single core

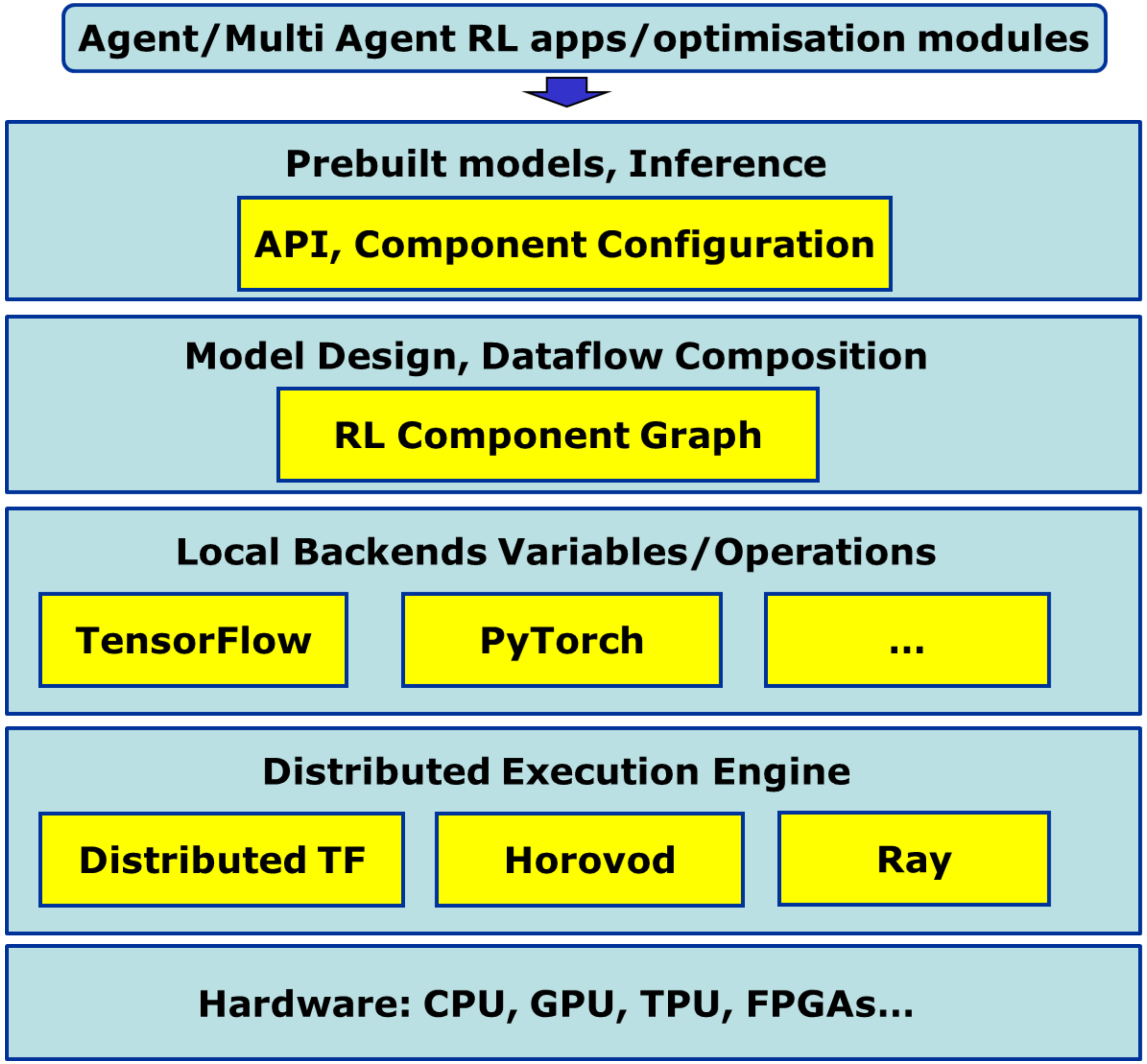
RL in Computer Systems: Practical Considerations

- Action spaces do not scale:
  - Systems problems often combinatorial
- Exploration in production system not a good idea
  - Unstable, unpredictable
- Simulations can oversimplify problem
  - Expensive to build, not justified versus gain
- Online steps take too long

Dynamic Control Flow in Current Frameworks

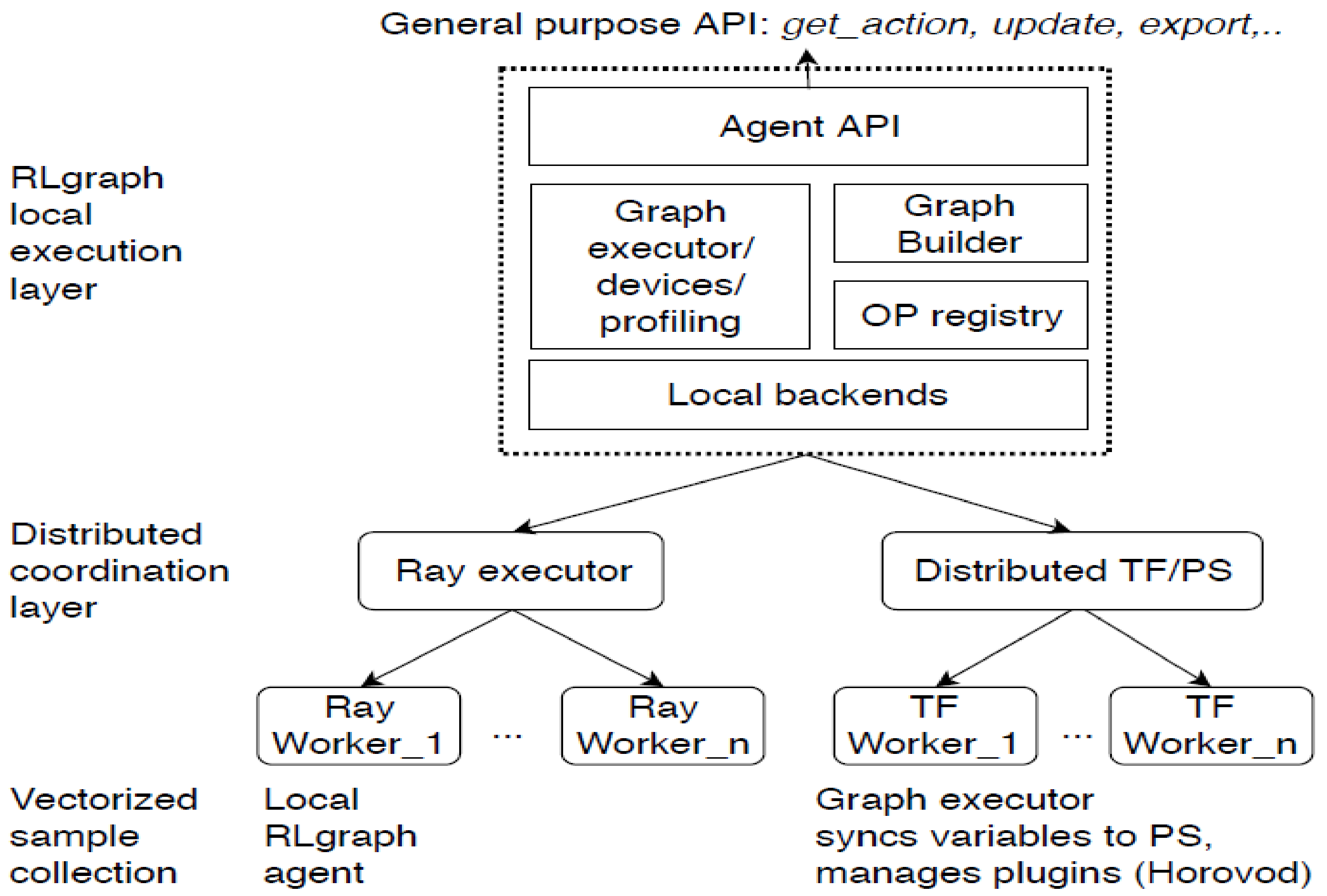
- There are static computation frameworks WITHOUT dynamic control flow (mxnet, cntk) -> dynamic control flow is in the out of graph host program.
- There are dynamic computation graph frameworks WITH dynamic control flow (PyTorch, DyNet) -> graph is only implicitly defined via imperative operations, cannot do static graph optimisations, typically slower but easier to use.
- There is static computation with dynamic differentiable control flow in the graph -> only TensorFlow offers this among modern deep learning frameworks.

RLgraph: Modular Dataflow Composition

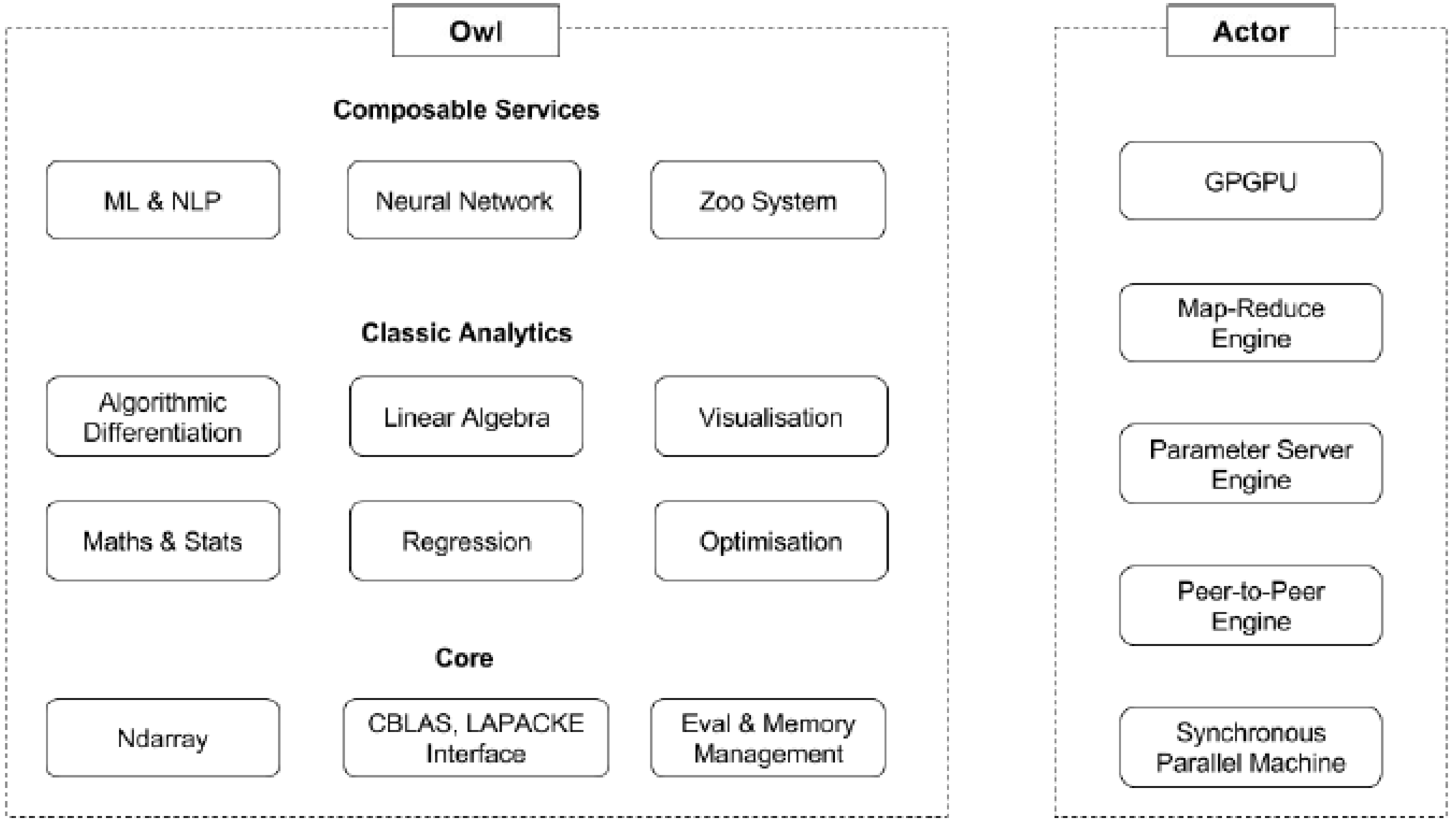


RLGraph: Separate Local and Distributed Execution

- High performance RL computation graphs for RL with different distributed backends



OWL Architecture for OCaml



Owl + Actor = Distributed & Parallel Analytics

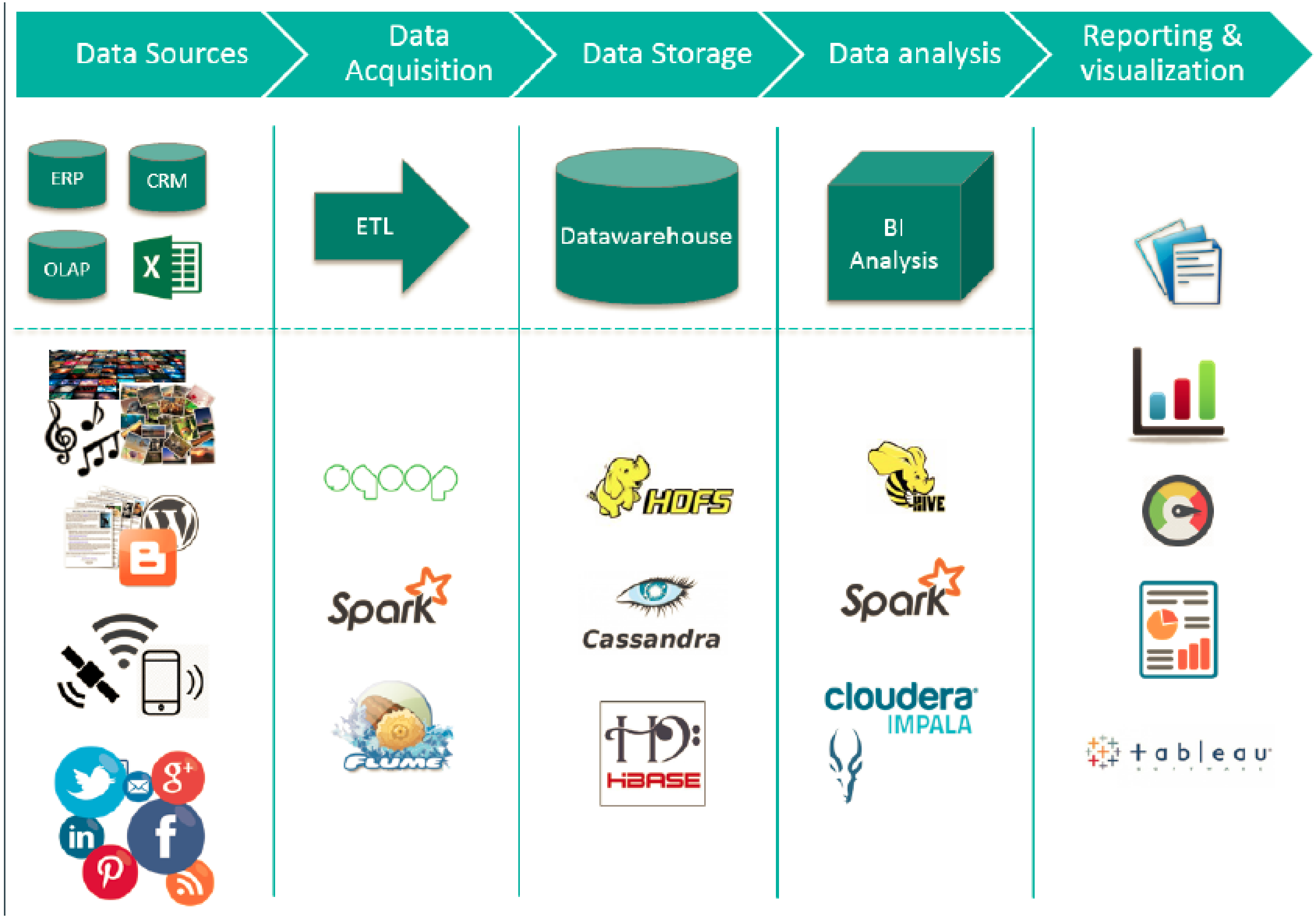
Owl provides numerical backend; whereas Actor implements the mechanisms of distributed and parallel computing. Two parts are connected with functors.

Various system backends allows us to write code once, then run it from cloud to edge devices, even in browsers.

Same code can run in both sequential and parallel mode with Actor engine.

By Liang Wang in 2018

Pipeline of Data Processing...



Modern Data Scientist: The sexiest job of 21th century

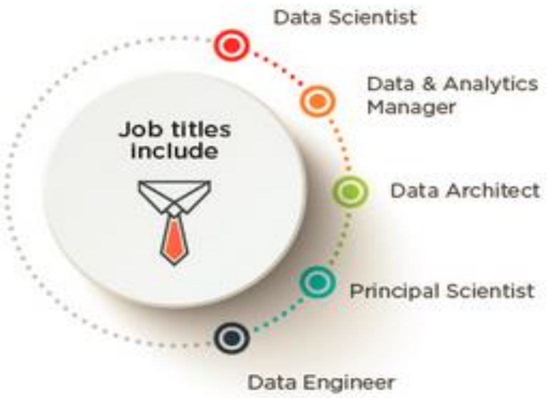
Many Courses offered: e.g. Master Certificate

- Data scientist is the pinnacle rank in an analytics organisation. You will be required to understand the business problem, design the analysis, collect and format the required data, apply algorithms or techniques using the correct tools, and finally make recommendations backed by data.

£ 1799

ENROLL NOW

* VAT Included



Course 1
Data Science with SAS Training

Course 2
Data Science Certification Training - R Programming

Course 3
Big Data Hadoop and Spark Developer

Course 4
Data Science with Python

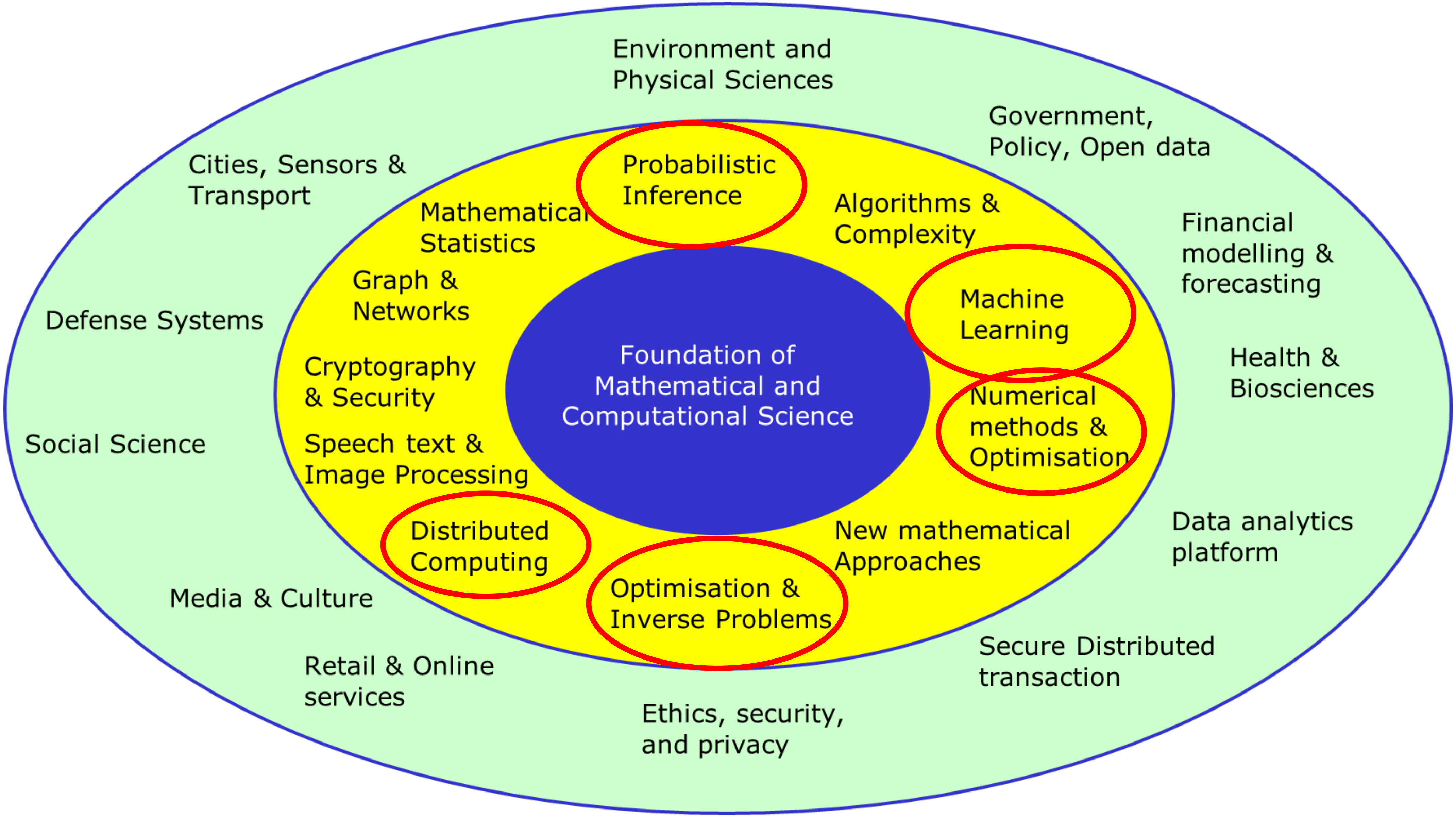
Course 5
Business Analytics with Excel

Course 6
Machine Learning

Course 7
Deep Learning with TensorFlow

Masters Certificate
*You will get individual certificates for each course.

Landscape of Data Science Research in ATI



AutoML: Neural Architecture Search

Current: ML expertise + Data + Computation

AutoML aims turning into: Data + 100X Computation

- Use of Reinforcement Learning, Evolutionary Algorithms

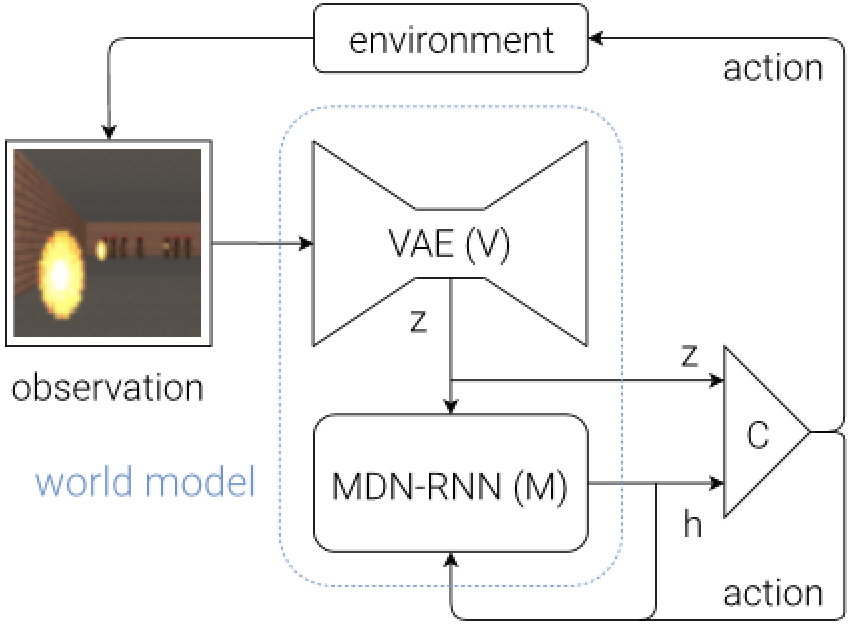
- ..and tune network model?
  - Graph transformation
  - Compression
  - + Hyper parameter tuning

RL Model Building

- Current: Simulation based if there is data

World Model:

- Training RL algorithms using variational auto encoders (simulator like)
  - Use randomly collected data as input and train to build a compact model
  - Train the compact model with RNN predict future steps the model, then evolve the controller to maximise the expected cumulative reward of roll out



Modern Theory of Deep Learning

Why does it work so well?

- On the Expressive Power of Deep Neural Networks PMLR 2017: understanding of how and why neural network architectures achieve their empirical successes is still lacking.
- Ali Rahimi's talk at NIPS(NIPS 2017 Test-of-time award presentation)
- Deep Learning works in Practice. But Does it work in Theory? By L. Hoang and R. Guerraoui. (https://arxiv.org/pdf/1801.10437.pdf)

- Understanding deep learning requires rethinking generalisation
- Fundamental theory behind the paradoxical effectiveness of deep learning. Still open research problem…

Gap between Research and Practice

Device Placement Optimization with Reinforcement Learning

Azalia Mirhoseini [* 1 2] Hieu Pham [* 1 2] Quoc V. Le [1] Benoit Steiner [1] Rasmus Larsen [1] Yuefeng Zhou [1]
Naveen Kumar [3] Mohammad Norouzi [1] Samy Bengio [1] Jeff Dean [1]

20H with 80GPUs!

Research opportunities ahead!

http://www.cl.cam.ac.uk/~ey204/

Acknowledgements: Michael Schaarschmidt (RL) and Valentin Dalibard (BOAT).