Learning Systems

Research at the Intersection of Machine Learning & Data Systems

Joseph E. Gonzalez
Asst. Professor, UC Berkeley
jegonzal@cs.berkeley.edu

How can **machine learning** techniques
be used **to address systems challenges**?

Learning Systems

How can **systems** techniques
be used **to address machine learning challenges**?

How can **machine learning** techniques
be used **to address systems challenges**?

Learning Systems

How can **systems** techniques
be used **to address machine learning challenges**?

How can **machine learning** techniques
be used **to address systems challenges**?

Systems are getting increasing complex:

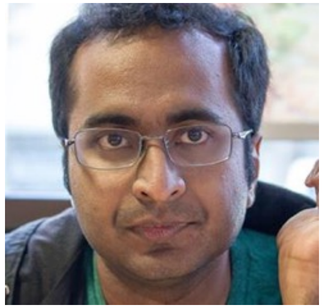
- ➢ Resource Disaggregation → growing diversity of system configurations and freedom to add resources as needed

- ➢ New Pricing Models → dynamic pricing and potential to bid for different types of resources

- ➢ Data-centric Workloads → performance depends on interaction between system, algorithms, and data

Paris
Performance Aware Runtime Inference System


Neeraja Yadwadkar


Bharath Hariharan

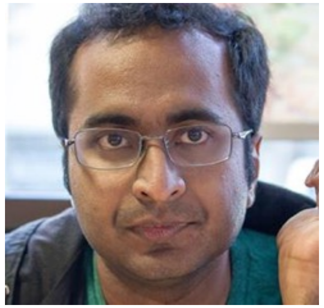

Randy Katz

➢ What vm-type should I use to run my experiment?

r3.2xlarge
m4.large
m4.4xlarge
r3.xlarge
c4.large
t2.large
r3.large
m3.xlarge
g2.8xlarge
m4.2xlarge
g2.2xlarge
m3.2xlarge
m3.medium
t2.micro
c4.xlarge
c4.2xlarge
m3.large
c4.4xlarge
r3.8xlarge
r3.4xlarge
c4.8xlarge
t2.small
t2.nano
m4.xlarge
x1.32xlarge
t2.medium
m4.10xlarge

Paris
Performance Aware Runtime Inference System

Neeraja Yadwadkar



Bharath Hariharan

Randy Katz

➢ What vm-type should I use to run my experiment?

r3.2xlarge xlarge m3.xlarg g2.2xlarge c4.large t2.large m4.large m4.4xlarge r3.xlarge r3.large m3.medium g2.8xlarge

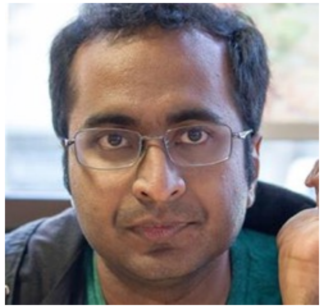
54 Instance Types

t2.micro m3.large r3.4xlarge m3.2xla r3.8xlarge c4.2xlarge c4.8xlarge c4.4xlarge c4.xlarge m4.xlarge t2.small t2.nano x1.32xlarge t2.medium m4.10xlarge

Paris

Performance Aware Runtime Inference System


Neeraja Yadwadkar


Bharath Hariharan


Randy Katz

- What vm-type should I use to run my experiment?

AWS 54

Azure 25

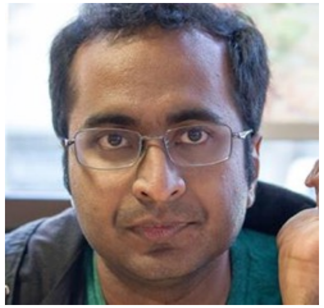
Google 18

- **Answer:** workload specific and depends on **cost** & **runtime** goals

Paris

Performance Aware Runtime Inference System

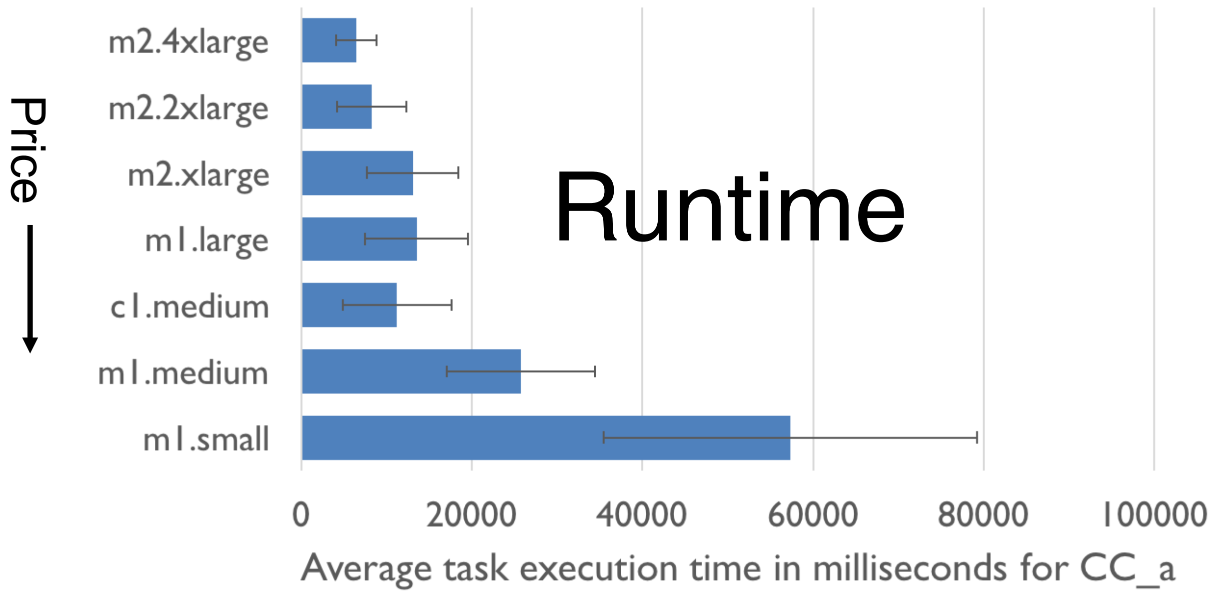
Neeraja Yadwadkar



Bharath Hariharan

Randy Katz

- ➢ Best vm-type depends on workload as well as **cost** & **runtime** goals



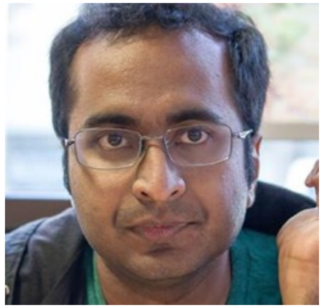
Which VM will cost me the least?

m1.small is cheapest?

Paris

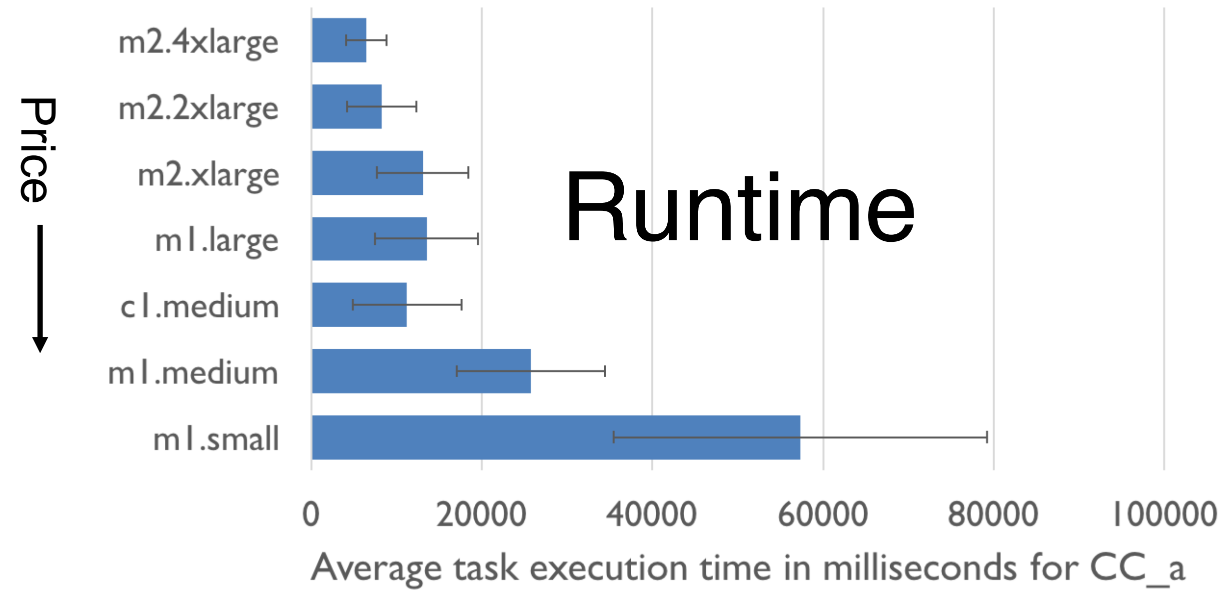
Performance Aware Runtime Inference System

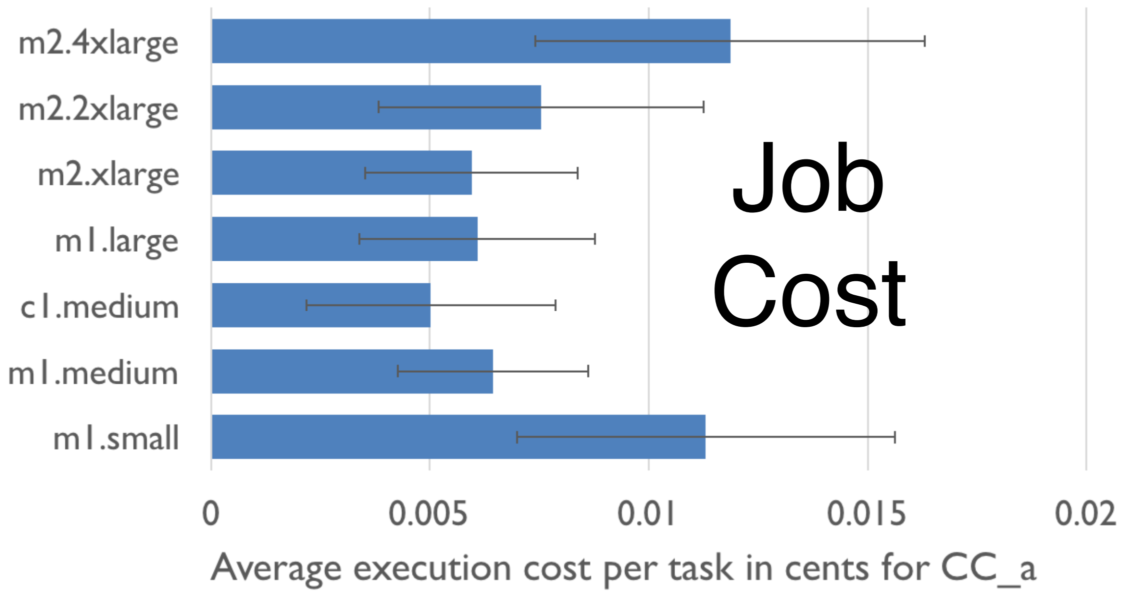

Neeraja Yadwadkar


Bharath Hariharan


Randy Katz

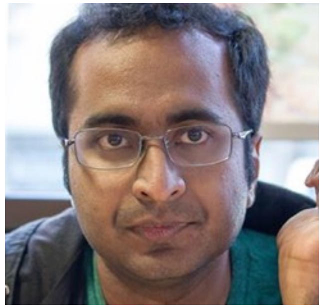
➢ Best vm-type depends on workload as well as **cost** & **runtime** goals





Requires accurate **runtime prediction.**

Paris
Performance Aware Runtime Inference System

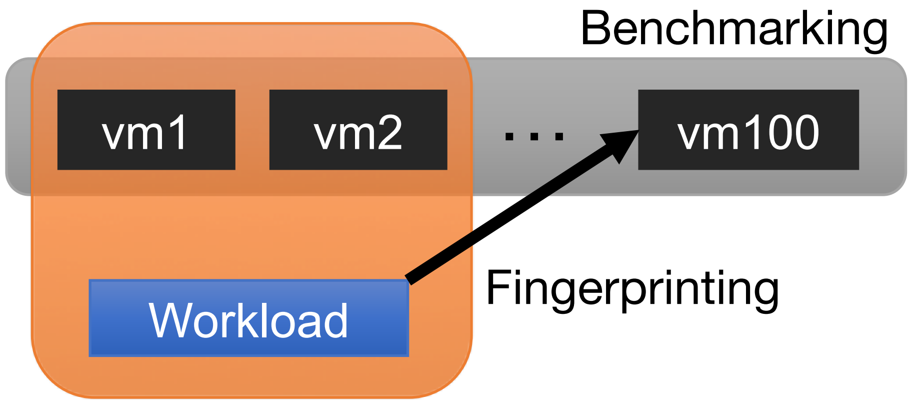




Neeraja Yadwadkar

Bharath Hariharan

Randy Katz

- **Goal:** Predict the runtime of ***workload w*** on ***VM type v***
  - **Challenge:** How do we model workloads and VM types
- ***Insight:***
  - Extensive *benchmarking* to model relationships between VM types
    - Costly but run once for all workloads
  - Lightweight workload *"fingerprinting"* by on a small set of test VMs
  - Generalize workload performance on other VMs
- **Results:** Runtime prediction 17% Relative RMSE (56% Baseline)



Hemingway*

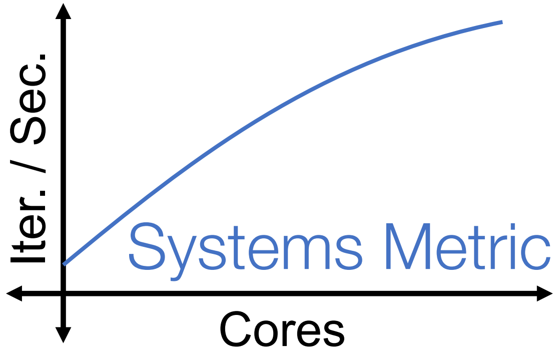
Modeling Throughput and Convergence for ML Workloads

Shivaram Venkataraman

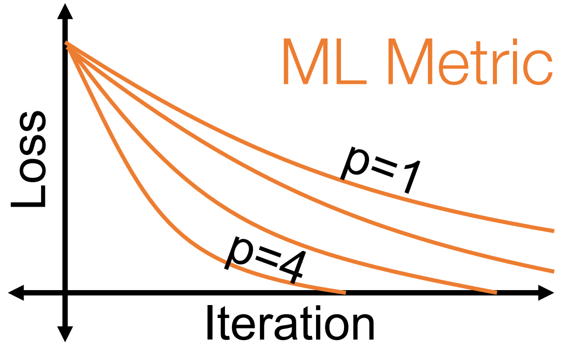
Xinghao Pan

Zi Zheng

- What is the best algorithm and level of parallelism for an ML task?
  - **Trade-off:** Parallelism, Coordination, & Convergence
- **Research challenge:** Can we model this trade-off explicitly?



$I(p)$ Iterations per second as a function of cores ***p***



$L(i,p)$ Loss as a function of iterations ***i*** and cores ***p***

We can estimate ***I*** from data on many systems

We can estimate ***L*** from data for our problem

*follow-up work to Shivaram's Ernest paper

Hemingway*

Modeling Throughput and Convergence for ML Workloads

Shivaram Venkataraman

Xinghao Pan

Zi Zheng

- What is the best algorithm and level of parallelism for an ML task?
  - **Trade-off:** Parallelism, Coordination, & Convergence
- **Research challenge:** Can we model this trade-off explicitly?

$L(i, p)$ Loss as a function of iterations ***i*** and cores ***p***

$I(p)$ Iterations per second as a function of cores ***p***

$$\mathbf{loss}(t, p) = L\left(t * I\left(p\right), p\right)$$

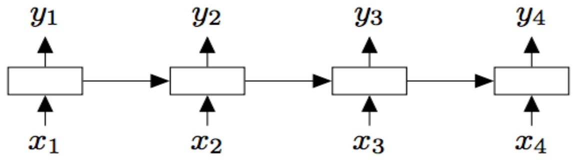
- How long does it take to get to a given loss?
- Given a time budget and number of cores which algorithm will give the best result?

*follow-up work to Shivaram's Ernest paper

Deep Code Completion

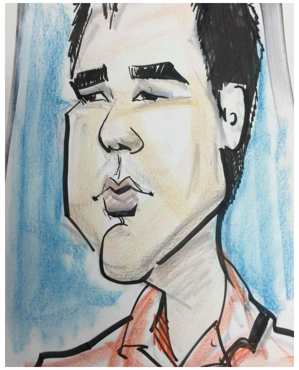
Neural architectures for reasoning about programs

- **Goals:**
  - Smart naming of variables and routines
  - Learn coding styles and patterns
  - Predict large code fragments
- Char and Symbol LSTMs



  - Programs are more tree shaped…

Xin Wang



Chang Liu

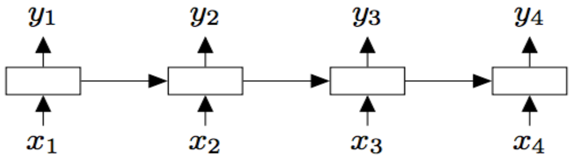
Dawn Song

```
def fib(x):
  if x < 2 :
     return x
  else:
     y = fib(x-1) + fib(x-2)
     return y
```

Deep Code Completion

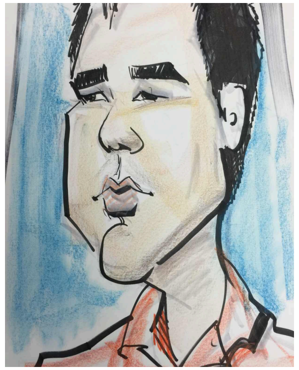
Neural architectures for reasoning about programs

- **Goals:**
  - Smart naming of variables and routines
  - Learn coding styles and patterns
  - Predict large code fragments
- Char and Symbol LSTMs



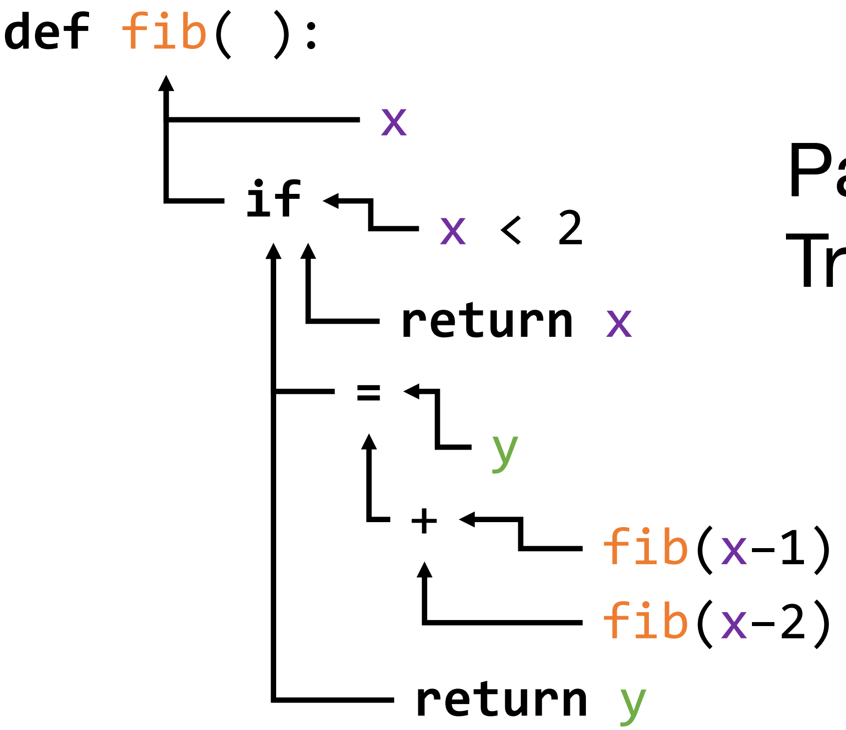
  - Programs are more tree shaped…

Xin Wang



Chang Liu

Dawn Song

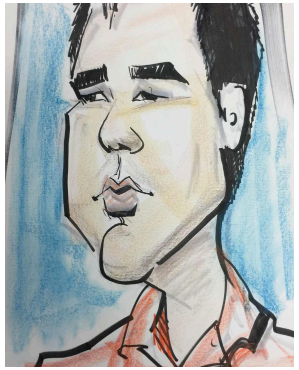


Parse Tree

Deep Code Completion

Neural architectures for reasoning about programs

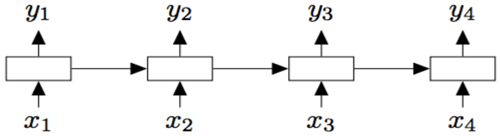
Xin Wang



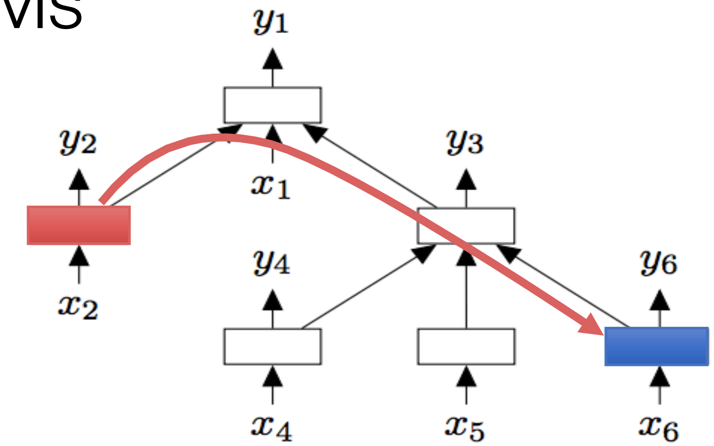
Chang Liu

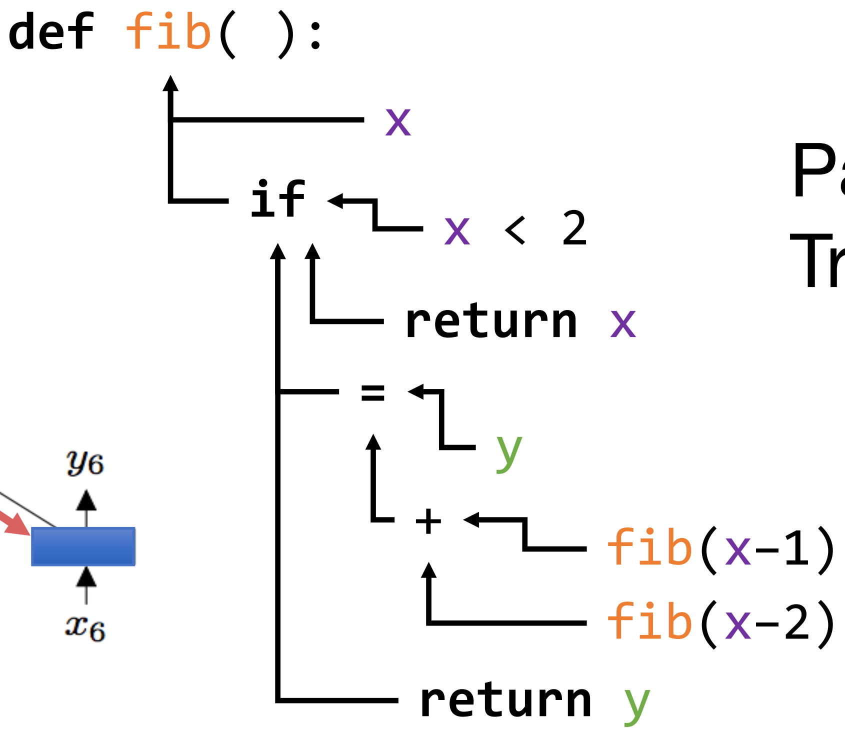
Dawn Song

- **Goals:**
  - Smart naming of variables and routines
  - Learn coding styles and patterns
  - Predict large code fragments
- Char and Symbol LSTMs



- Exploring Tree LSTMs
  - Issue: dependencies flow in both directions





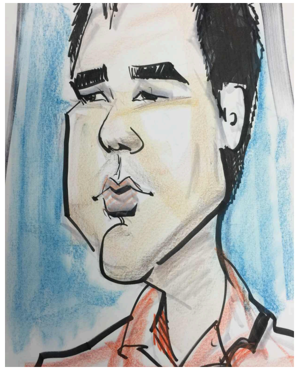
Parse Tree

Kai Sheng Tai, Richard Socher, Christopher D. Manning. *"Improved Semantic Representations From Tree-Structured Long Short-Term Memory Networks."* (ACL 2015)

Deep Code Completion

Neural architectures for reasoning about computer programs

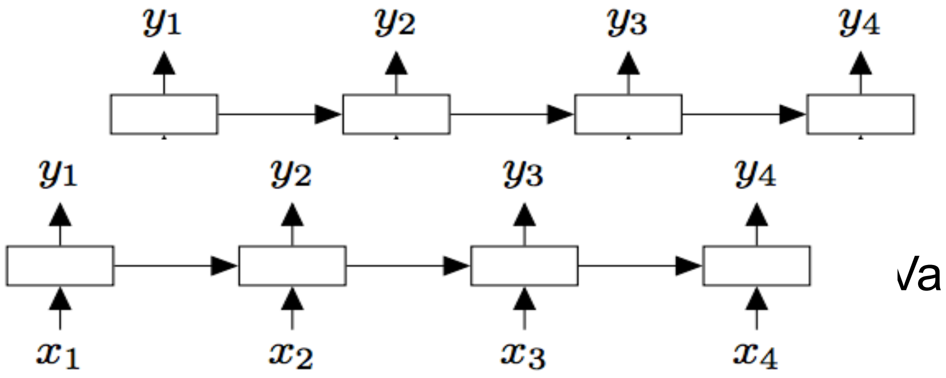
Xin
Wang



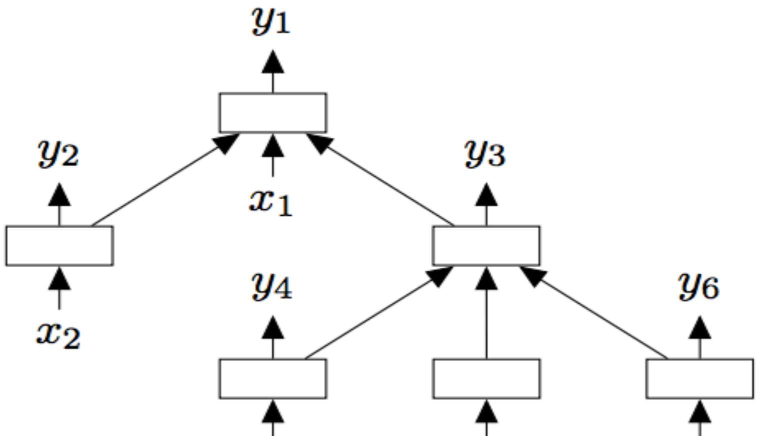
Chang
Liu

Dawn
Song

- **Goals:**
  - Smart naming of variables and routines
  - Learn coding styles and patterns
  - Predict large code fragments
- Current studying Char-LSTM and Tree-LSTM on benchmark C++ code and JavaScript code.
- Plan to extend Tree-LSTM with downward information flow



Vanilla LSTM



Tree- LSTM

Fun Code Sample Generated by Char-LSTM

Code Prefix

```
vector<string> words;
    vector<set<int> > paths;
    unordered_map<string, int> dict_map;
    dict.insert(start);
    dict.insert(end);
    int i = 0;
    for (unordered_set<string>::iterator iter = dict.begin(); iter != dict.end(); iter ++ , i++) {
        words.push_back(* iter);
        dict_map.insert(pair<string, int>(* iter, i));
        paths.push_back(set<int>());
    }
    vector<vector<int> > map;
    vector<int> distance;
    deque<int> queue;
    this->prepare_map(dict, dict_map, words, map, distance);
    int start_index = dict_map.find(start)->second;
    int end_index = dict_map.find(end)->second;
    distance[start_index] = 1;
    queue.push_back(start_index);
    while (!queue.empty())
        {
            int n = queue.front();
            for (int i = 0; i < map[n].size(); ++i) {
                if ((distance[n] + 1) < distance[map[n][i]]) {
```

Generated Code Sample

```
                queue[neigh] = 1 , path.push_back(i + 1) , group_dict[now] = opblue;
                    permutation.push_back(idxChack[N - start + 1]);
                    continue;
                } else {
                    minpi_crange = true;
                    path.pop_back();
                }
            }
            result.push_back(RowNum(next, vase - pres[start][2]));
        }
        return result;
    }
```

For now, the neural network can learn some code patterns like matching the parenthesis, if-else block, etc but the variable name issue still hasn't been solved.

*this is trained on the leetcode OJ code submissions from Github.

How can **machine learning** techniques
be used **to address systems challenges**?

How can **systems** techniques
be used **to address machine learning challenges**?

How can **machine learning** techniques
be used **to address systems challenges**?

How can **systems** techniques
be used **to address machine learning challenges**?

Systems for Machine Learning

Timescale: minutes to days

Systems: offline and batch optimized

Heavily studied ... primary focus of the ***ML research***

Please make a Logo!

DMLC
TensorFlow
Caffe
Temgine
GraphLab
APACHE Spark
MLbase
KeystoneML
GraphX
Splash
CoCoA
Please make a Logo!

Temgine
A Scalable Multivariate Time Series Analysis Engine

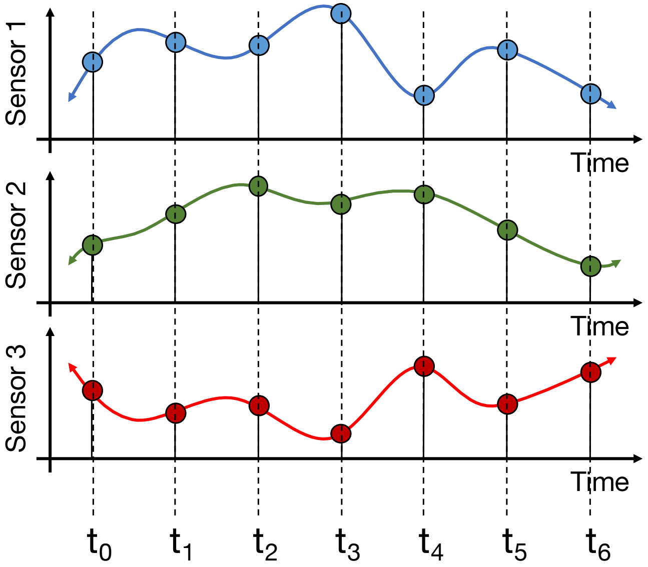
Francois Billetti
Evan Sparks
Xin Wang

Challenge:

- Estimate second order statistics
  - E.g. Auto-correlation, auto-regressive models, …
- for **high-dimensional** & **irregularly sampled** time series

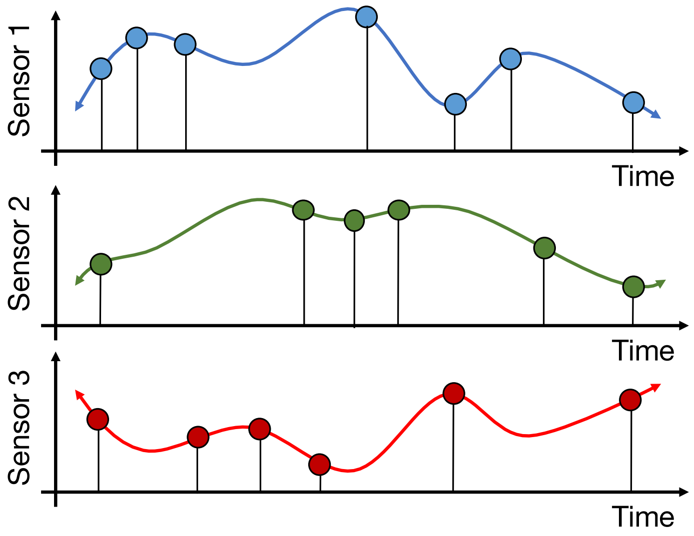
Regularly Sampled

Samples are easy to align (requires sorting)



Irregularly Sampled

Difficult to align!



Temgine

A Scalable Multivariate Time Series Analysis Engine



Francois Billetti

Evan Sparks

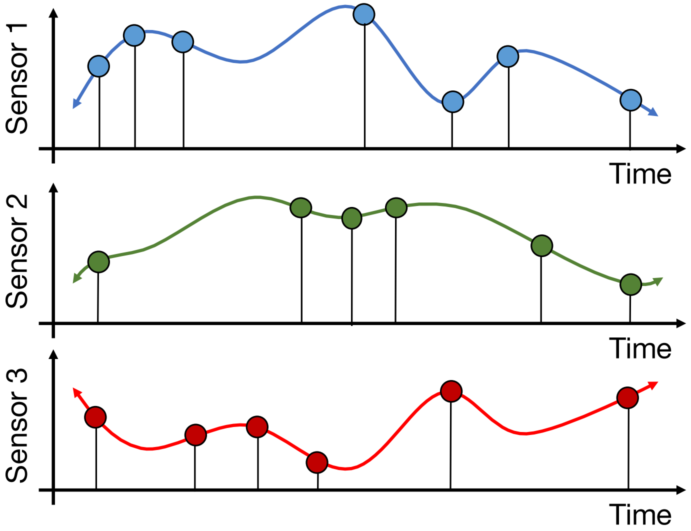
Xin Wang

Challenge:

- Estimate second order statistics
  - E.g. Auto-correlation, auto-regressive models, …
- for **high-dimensional** & **irregularly sampled** time series

Irregularly Sampled

Difficult to align!



Solution:

- Project onto Fourier basis
  - does not require data alignment
- Infer statistics in frequency domain
  - equivalent to kernel smoothing
  - analysis of bias – variance tradeoff

Temgine

A Scalable Multivariate Time Series Analysis Engine

Francois Billetti

Evan Sparks

Xin Wang

Challenge:

- ➢ Estimate second order statistics
  - ➢ E.g. Auto-correlation, auto-regressive models, …
- ➢ for **high-dimensional** & **irregularly sampled** time series

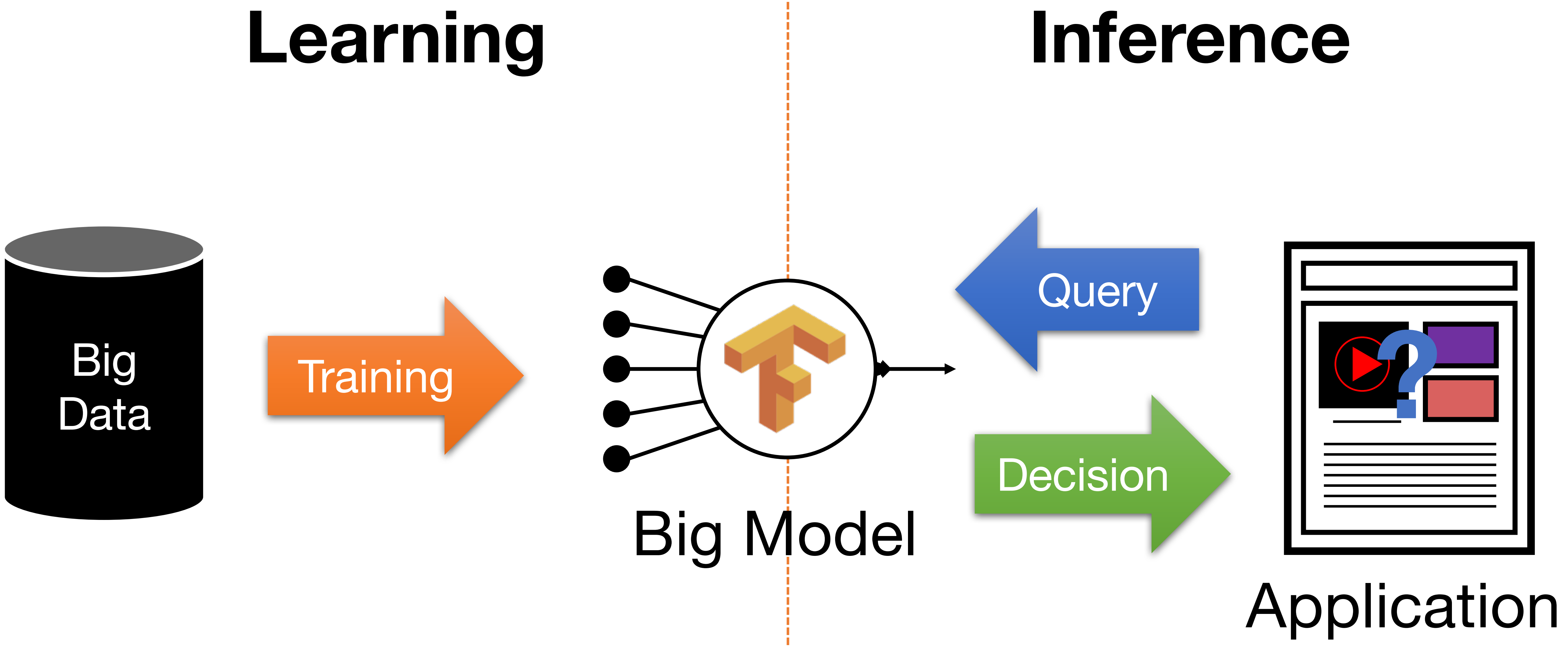
Solution:

- Project onto Fourier basis
  - does not require data alignment
- Infer statistics in frequency domain
  - equivalent to kernel smoothing
  - analysis of bias – variance tradeoff

Temgine

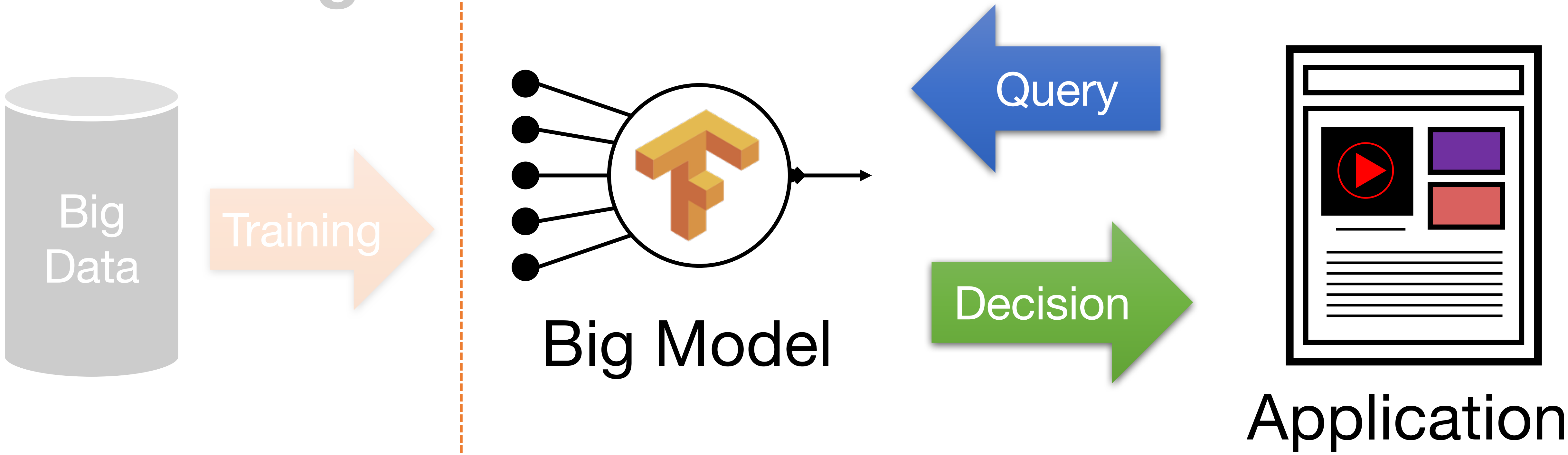
Define an operator DAG (like TF) and then rely on query-optimization to define efficient execution.

Learning


Learning
Inference
Big
Data
Training
Big Model
Query
Decision
Application

Learning

Inference

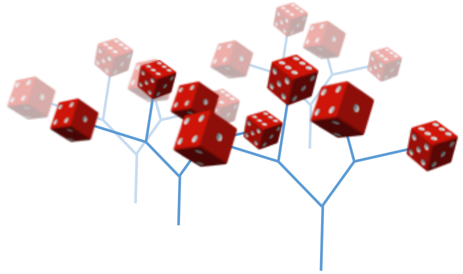


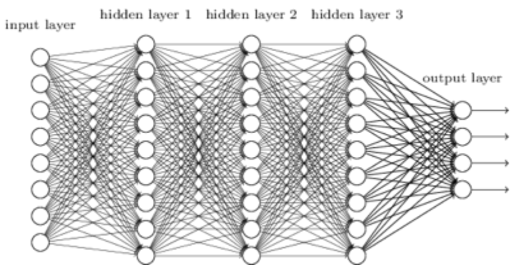
Timescale: ~10 milliseconds
Systems: *online* and *latency* optimized
Less Studied …

why is **Inference** challenging?

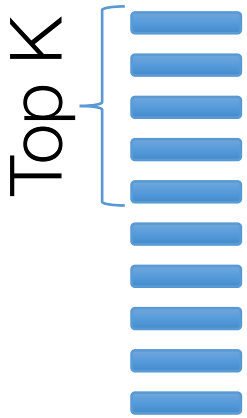
Need to render **low latency** (< 10ms) predictions for **complex**

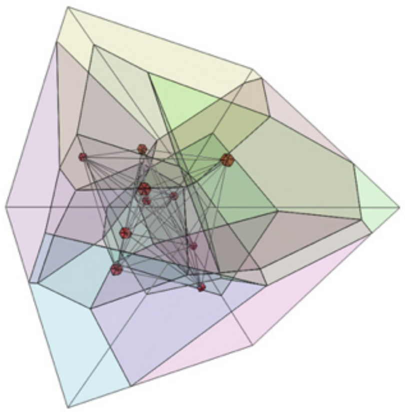
Models





Queries





Features

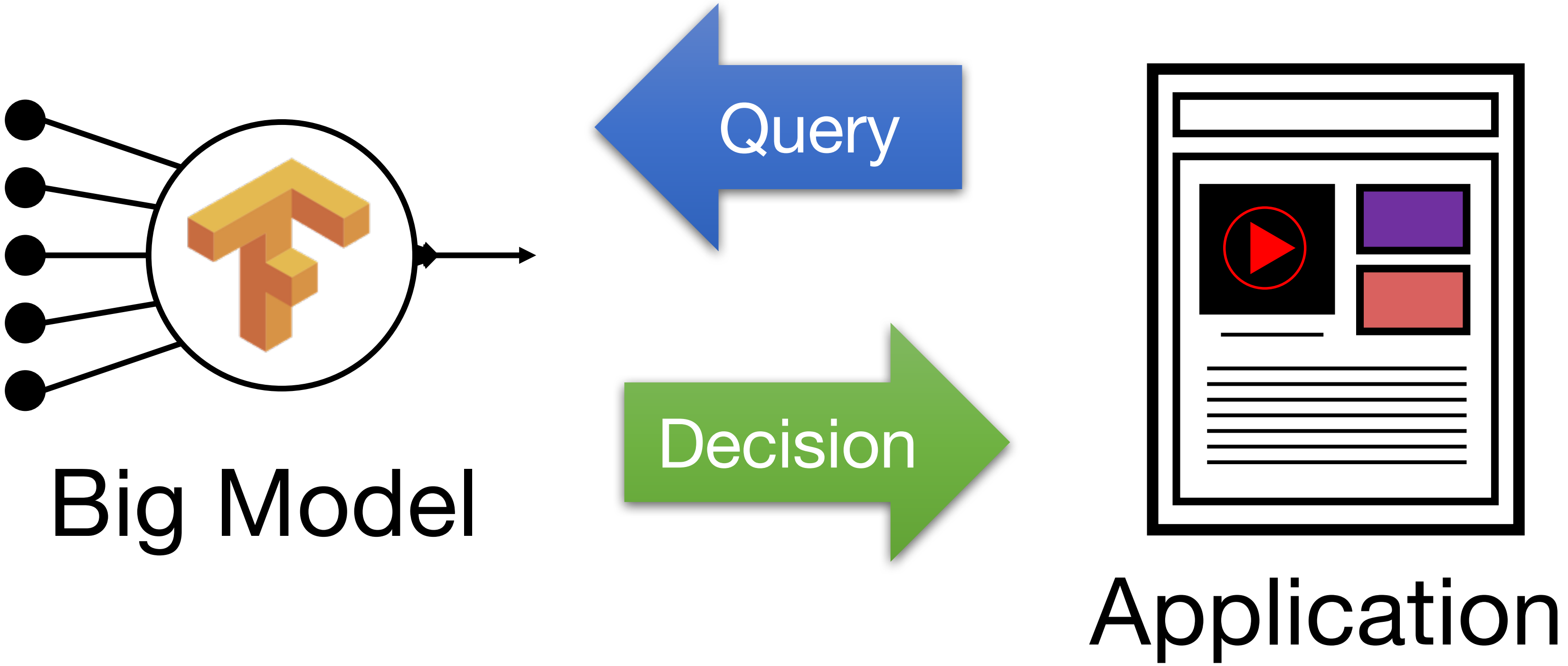
SELECT * FROM
users JOIN items,
click_logs, pages
WHERE …

under **heavy load** with system **failures**.

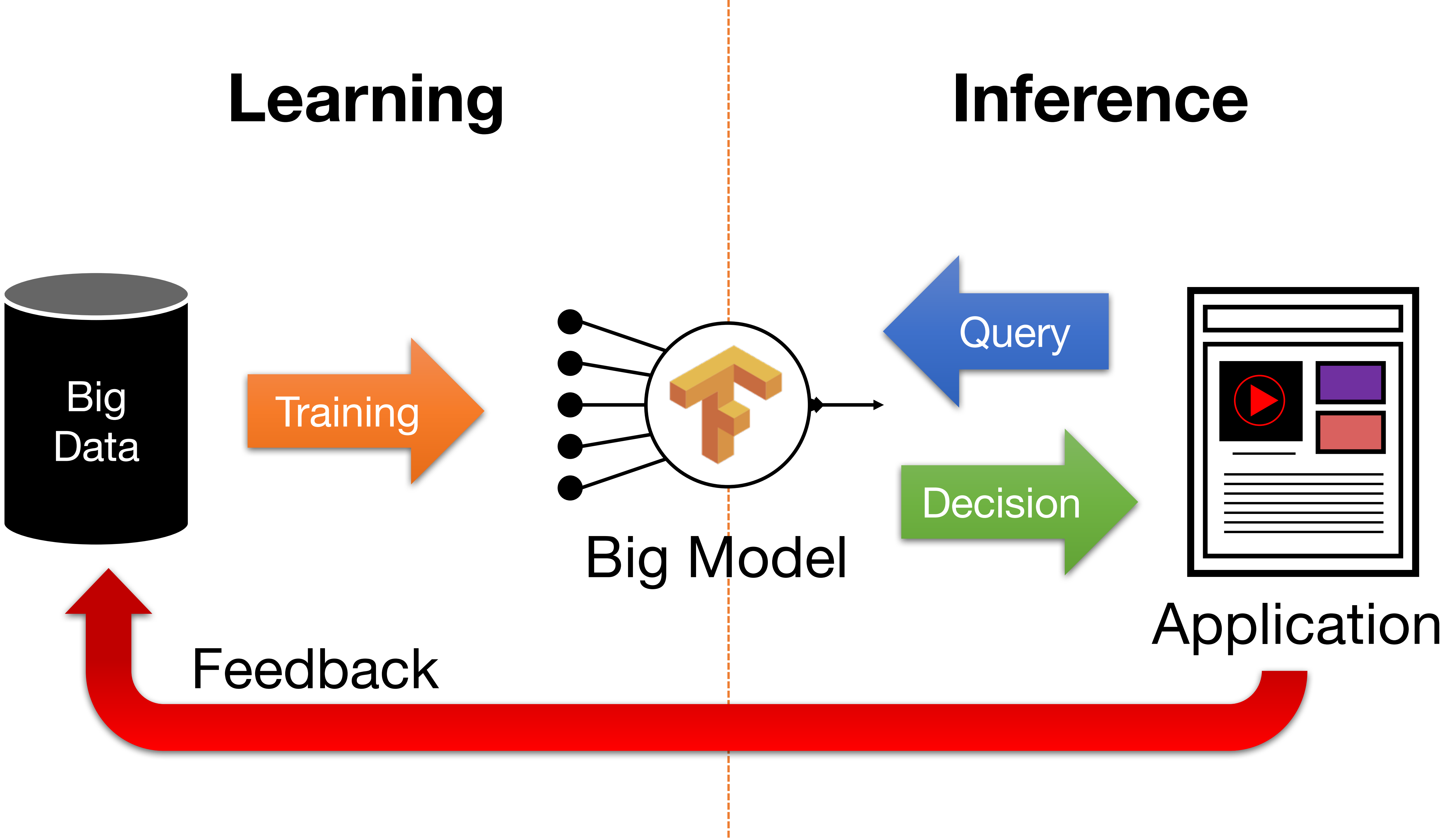
Learning

Claim:
next big area of research in scalable ML systems

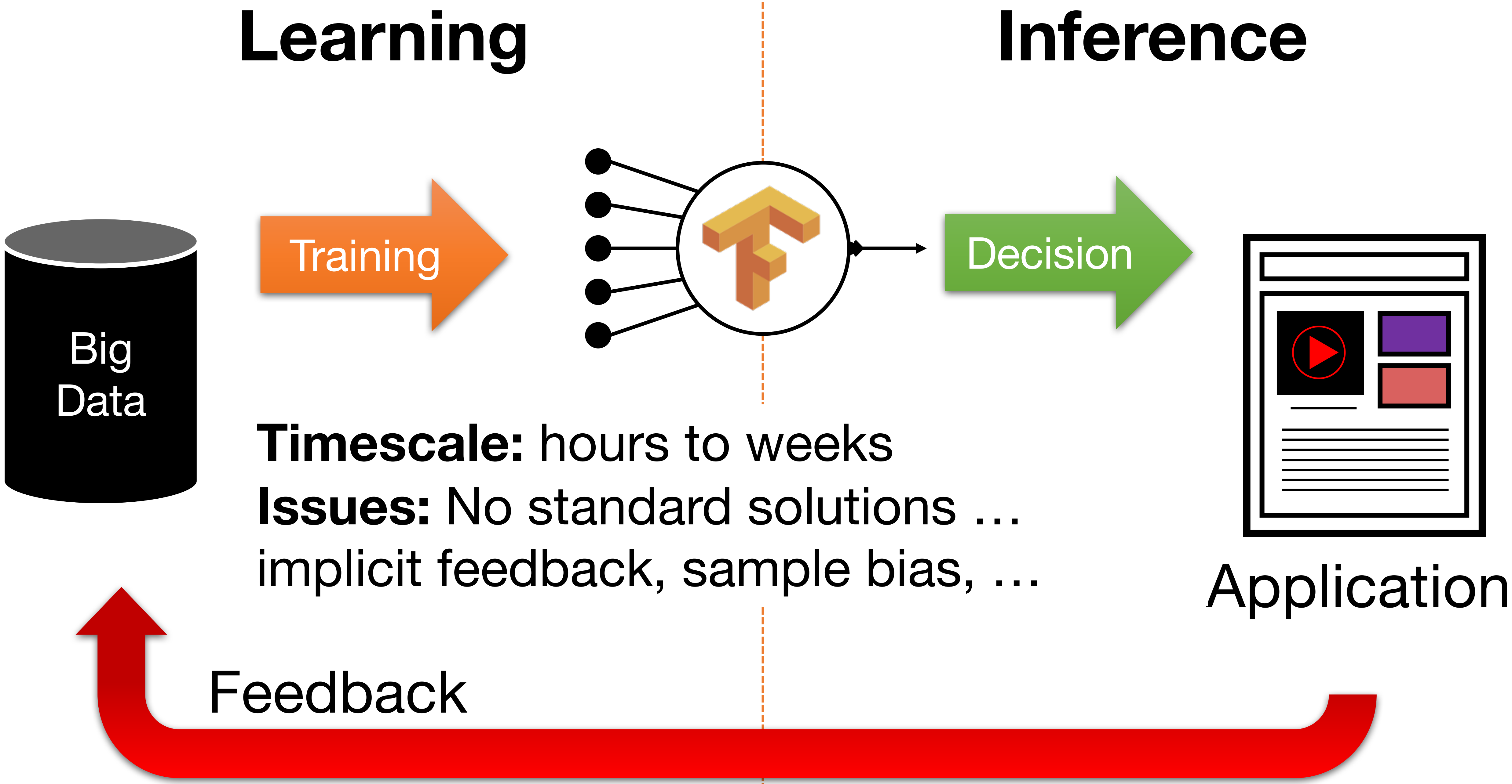
Inference



Timescale: ~10 milliseconds
Systems: *online* and *latency* optimized
Less studied …

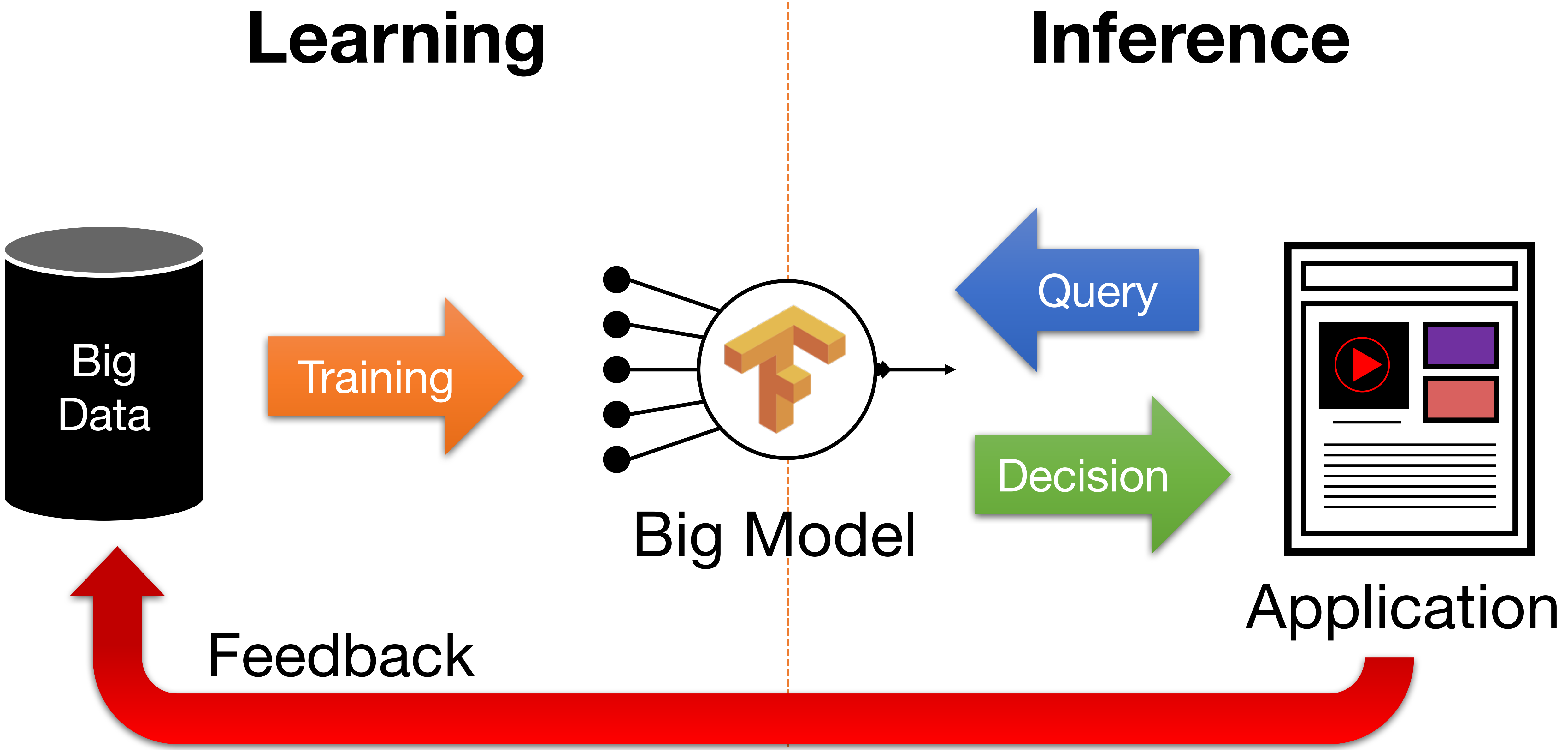


Learning
Inference
Big Data
Training
Big Model
Query
Decision
Application
Feedback


Learning
Inference
Big Data
Training
Decision
Application
Timescale: hours to weeks
Issues: No standard solutions …
implicit feedback, sample bias, …
Feedback

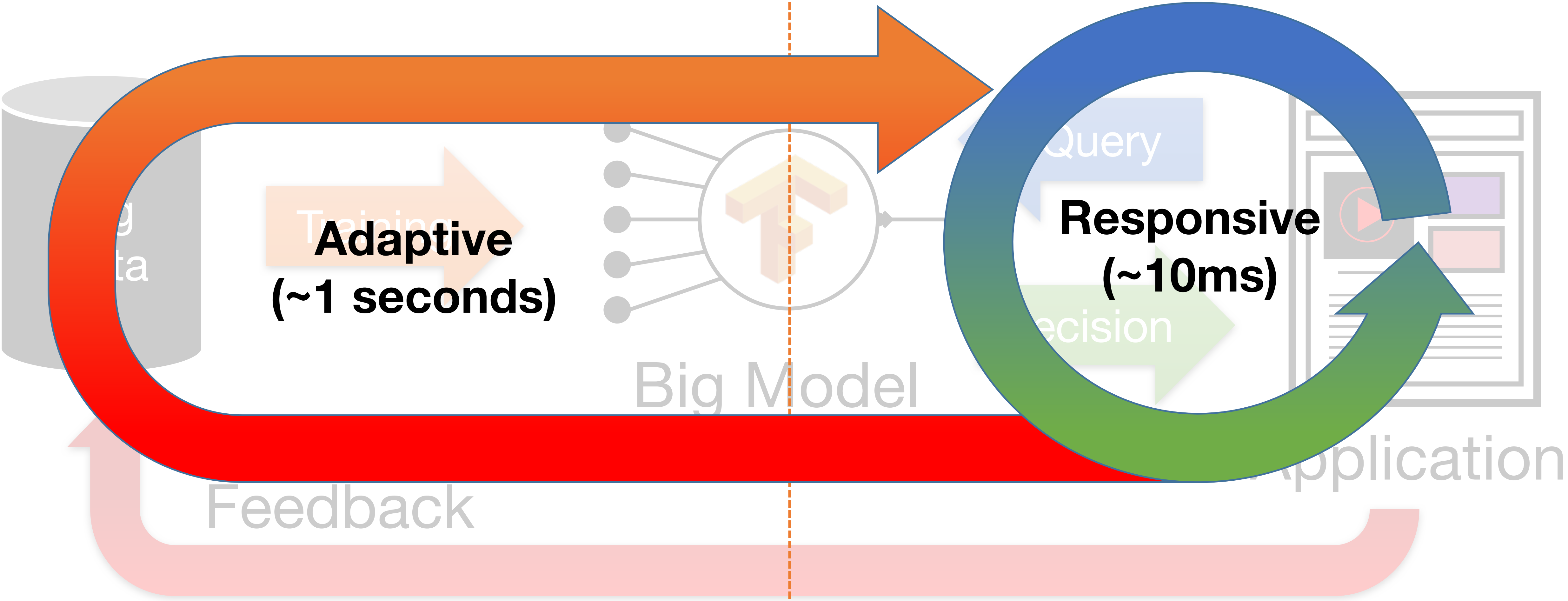
Why is Feedback challenging?

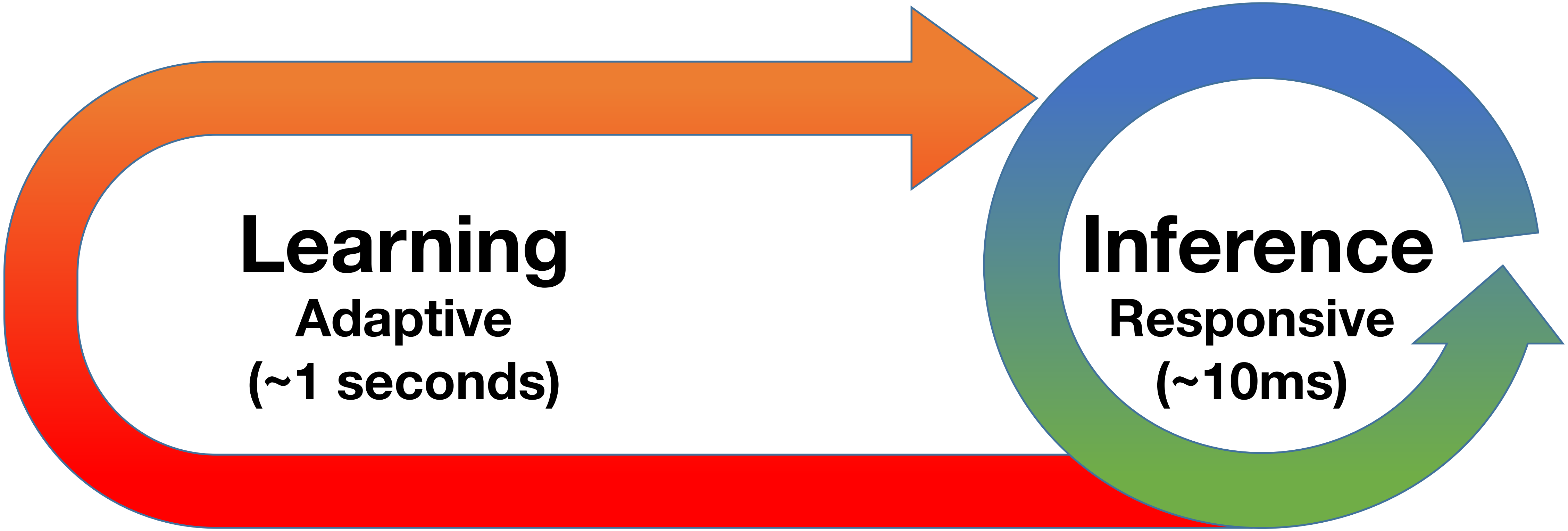
- Exposes system to **feedback loops**
  - Address Explore – Exploit trade-off in real-time
- Adverserial feedback
  - Opportunities for **multi-task learning** and **anomly detection**
- Need to address **temporal variation**
  - Need to model time directly? When do we forget the past?


Learning
Inference
Big
Data
Training
Big Model
Query
Decision
Application
Feedback

Learning

Inference



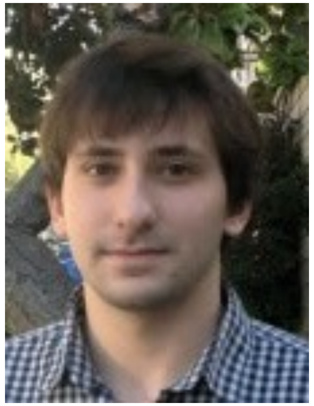


Techniques we are studying (or **should be** …):

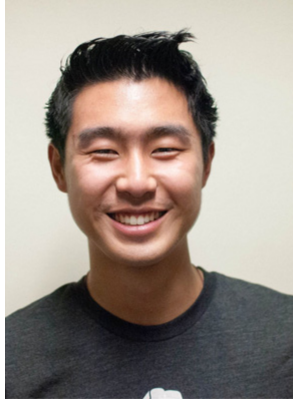
Multi-task Learning | Adaptive Batching | Approx. Caching | Anytime Inference | Model Switching | Meta-Policy RL

Online Ensemble Learning | Load Shedding | Model Compression | Inference on the Edge

Prediction Serving

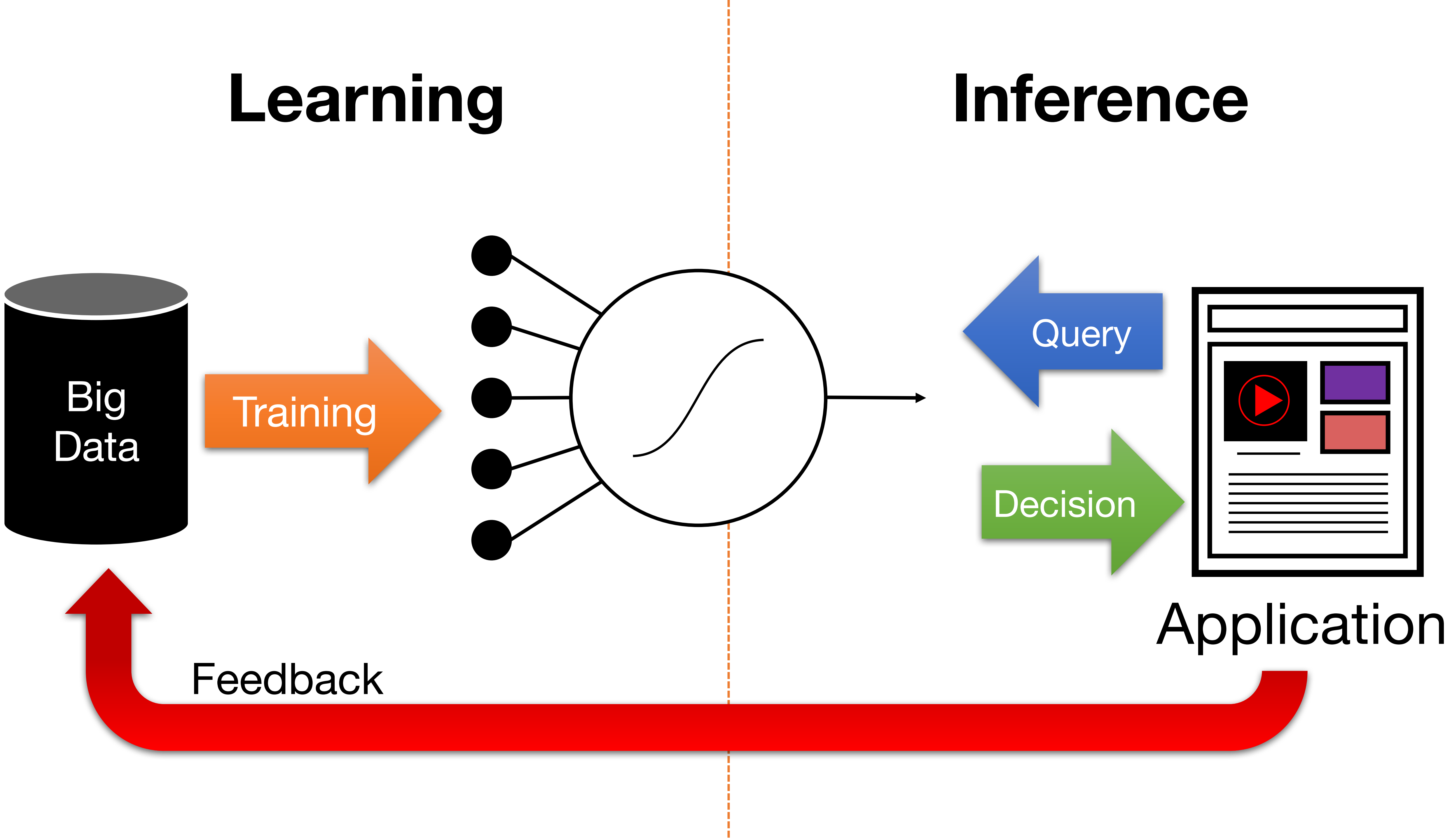

Daniel Crankshaw

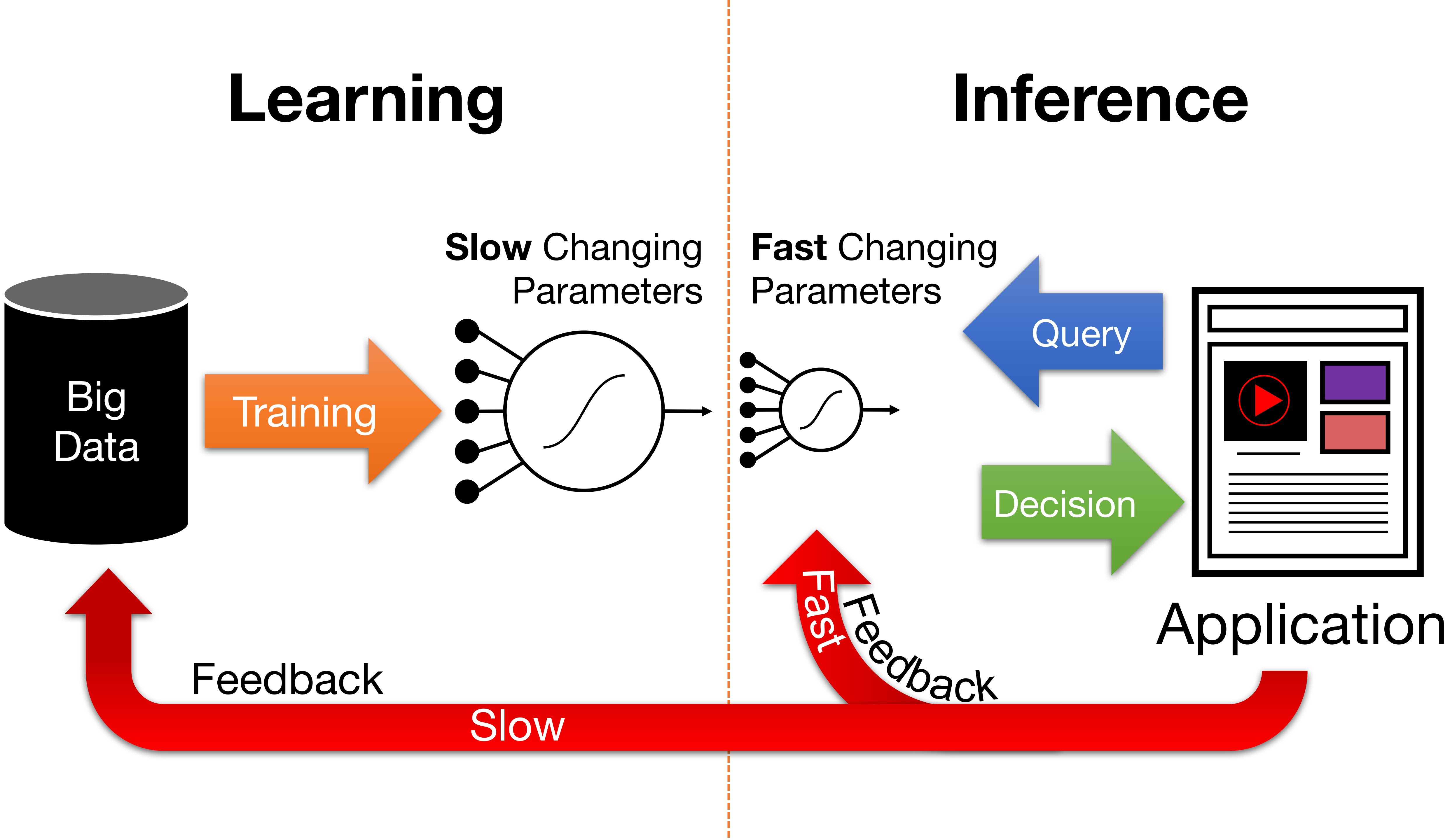

Xin Wang


Giulio Zhou


Michael Franklin


Ion Stoica


Learning
Inference
Big Data
Training
Query
Decision
Application
Feedback



Learning
Inference
Slow Changing Parameters
Fast Changing Parameters
Big Data
Training
Query
Decision
Application
Fast Feedback
Feedback
Slow

Hybrid Offline + Online Learning

Update feature functions **offline** using batch solvers
- Leverage high-throughput systems (Tensor Flow)
- Exploit slow change in population statistics

$$f(x;\theta)^T w_u$$

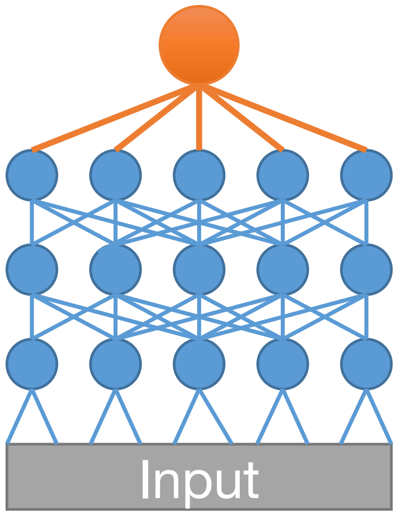
Update the user weights **online**:
- Simple to train + more robust model
- Address rapidly changing user statistics

Common modeling structure

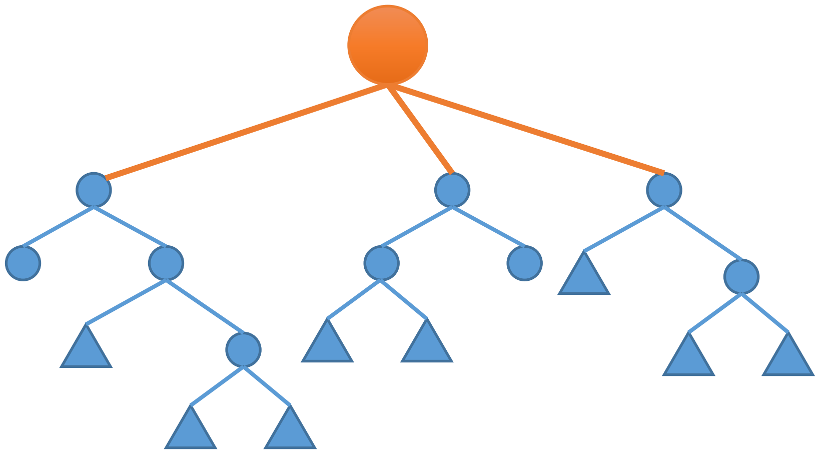
$$f(x;\theta)^T w_u$$

Matrix Factorization

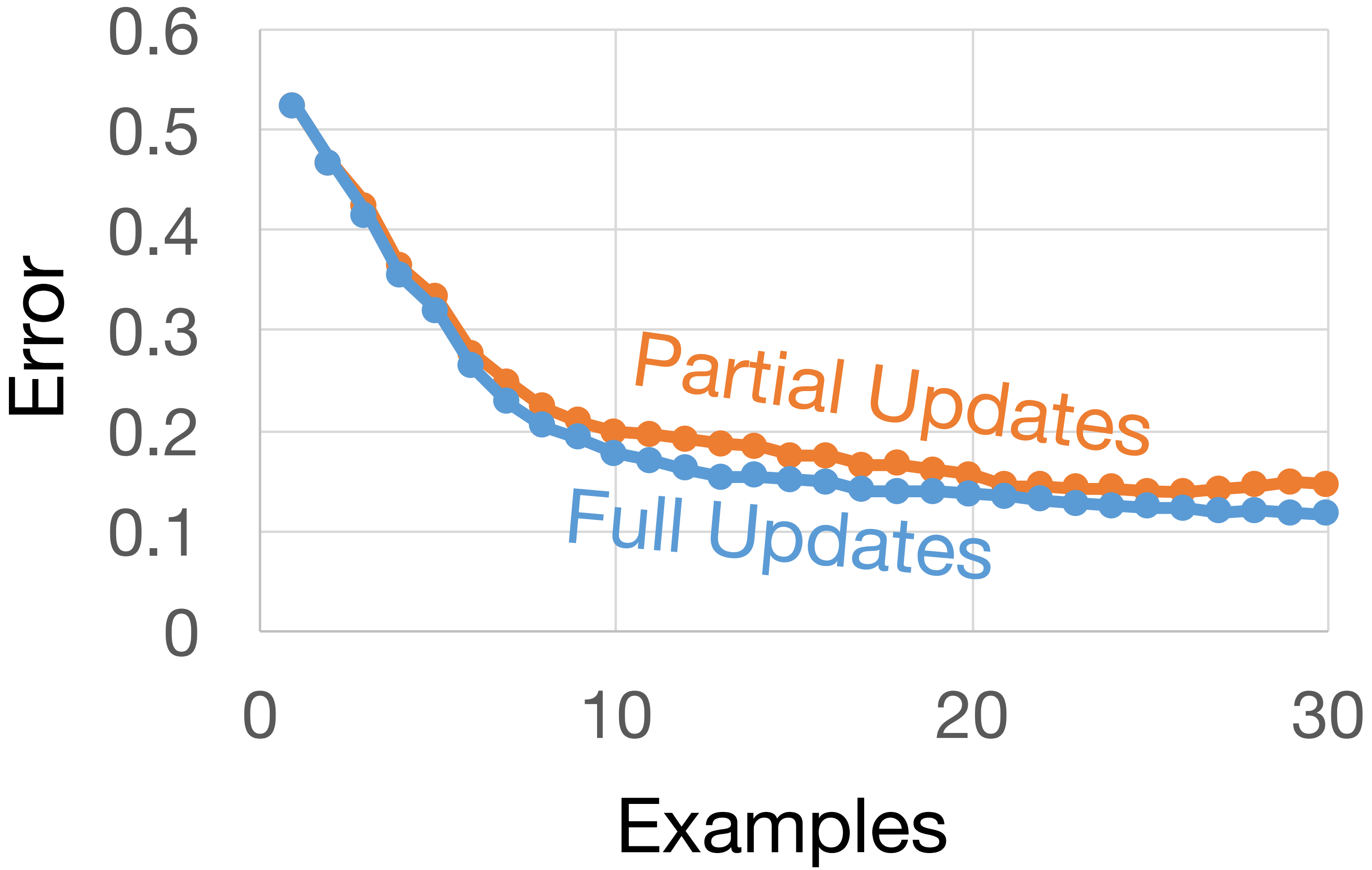
Deep Learning



Ensemble Methods

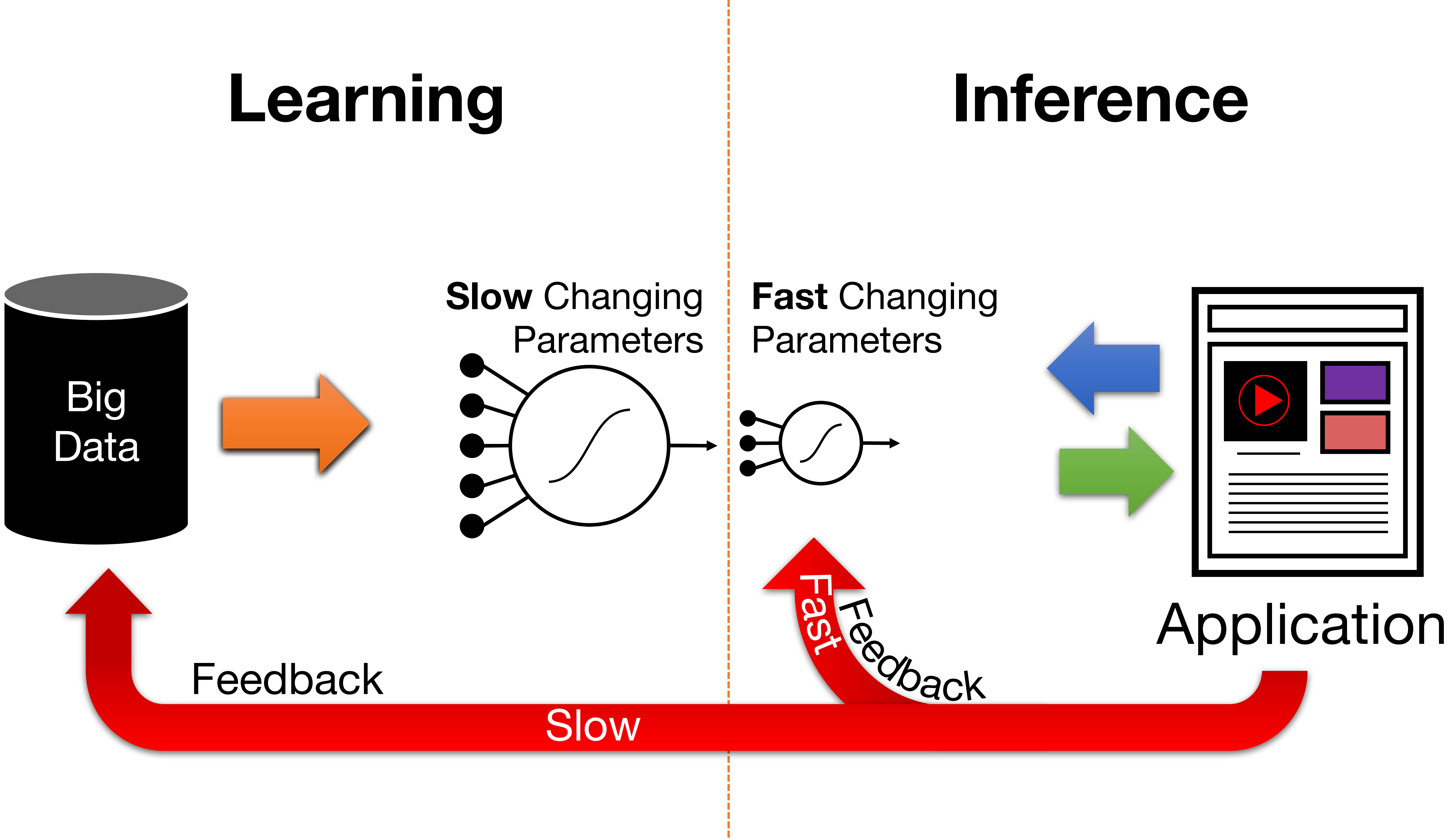


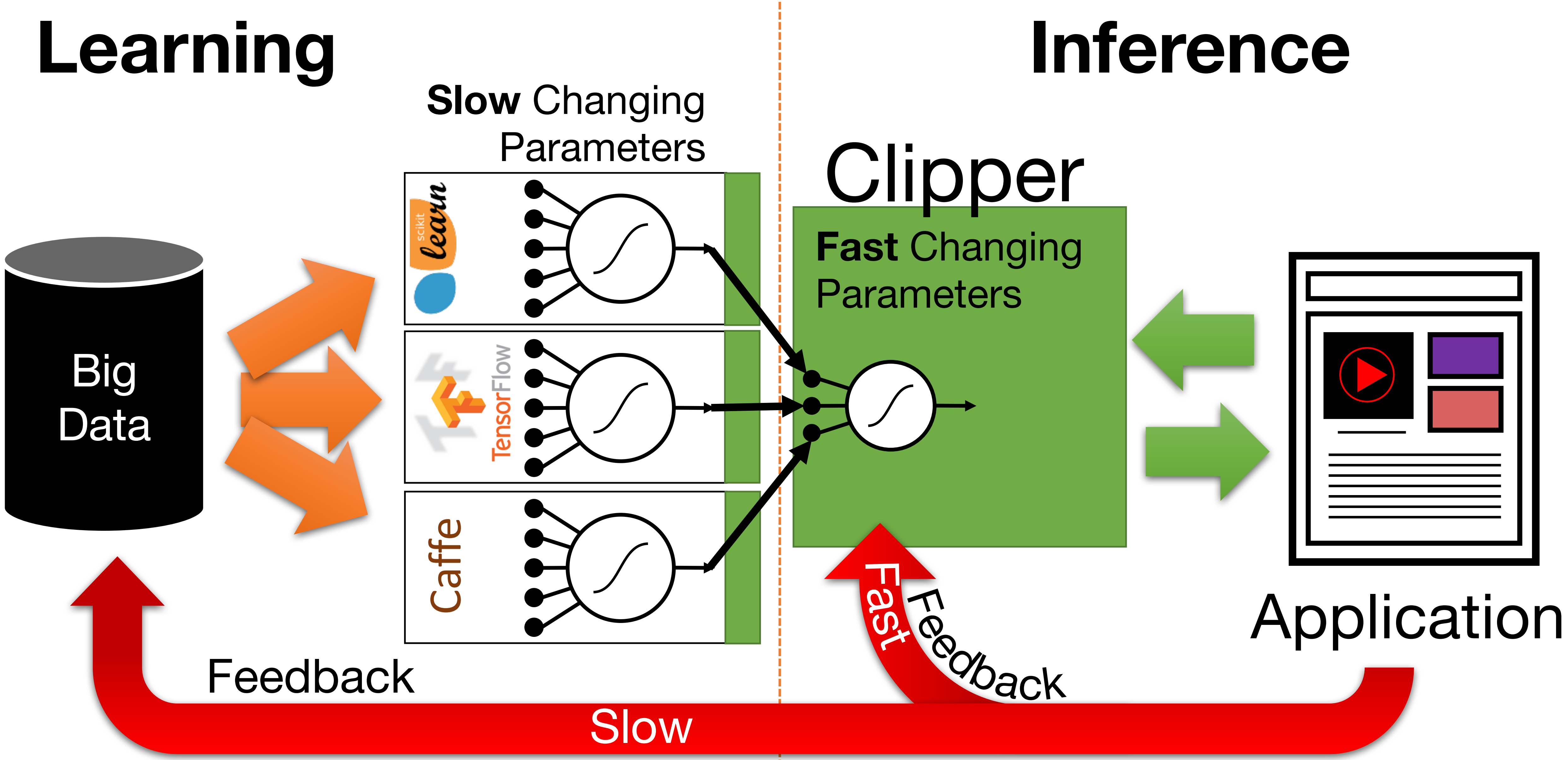
Clipper Online Learning for Recommendations
(Simulated News Rec.)



Partial Updates: *0.4 ms*
Retraining: *7.1 seconds*

>4 orders-of-magnitude
faster adaptation


Learning
Inference
Big Data
Slow Changing Parameters
Fast Changing Parameters
Application
Feedback
Slow
Fast
Feedback



Learning
Inference
Slow Changing Parameters
Clipper
Fast Changing Parameters
Big Data
TensorFlow
Caffe
Application
Feedback
Slow
Fast
Feedback

Clipper Serves Predictions across ML Frameworks

Fraud Detection

Content Rec.

NETFLIX

Personal Asst.

Robotic Control

Machine Translation

Clipper

theano

Dato Create

Caffe

TensorFlow

scikit learn

dmlc mxnet

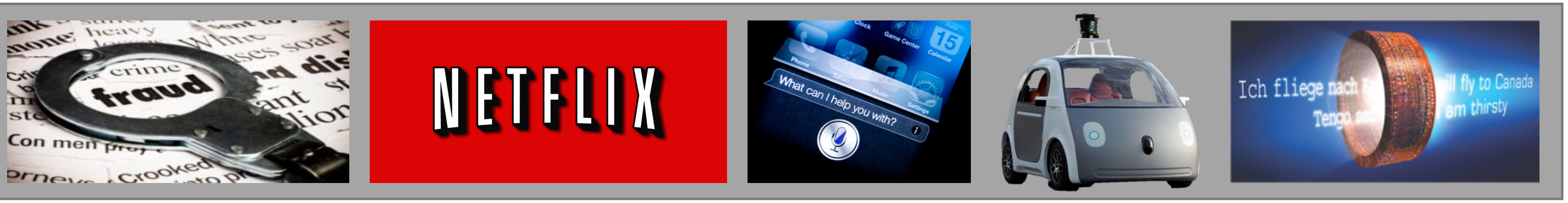
KeystoneML

VW

KALDI

Clipper Architecture

Applications



Predict ↕ **RPC/REST Interface** ↕ Observe

Clipper

theano

Dato Create

KeystoneML

Caffe

TensorFlow

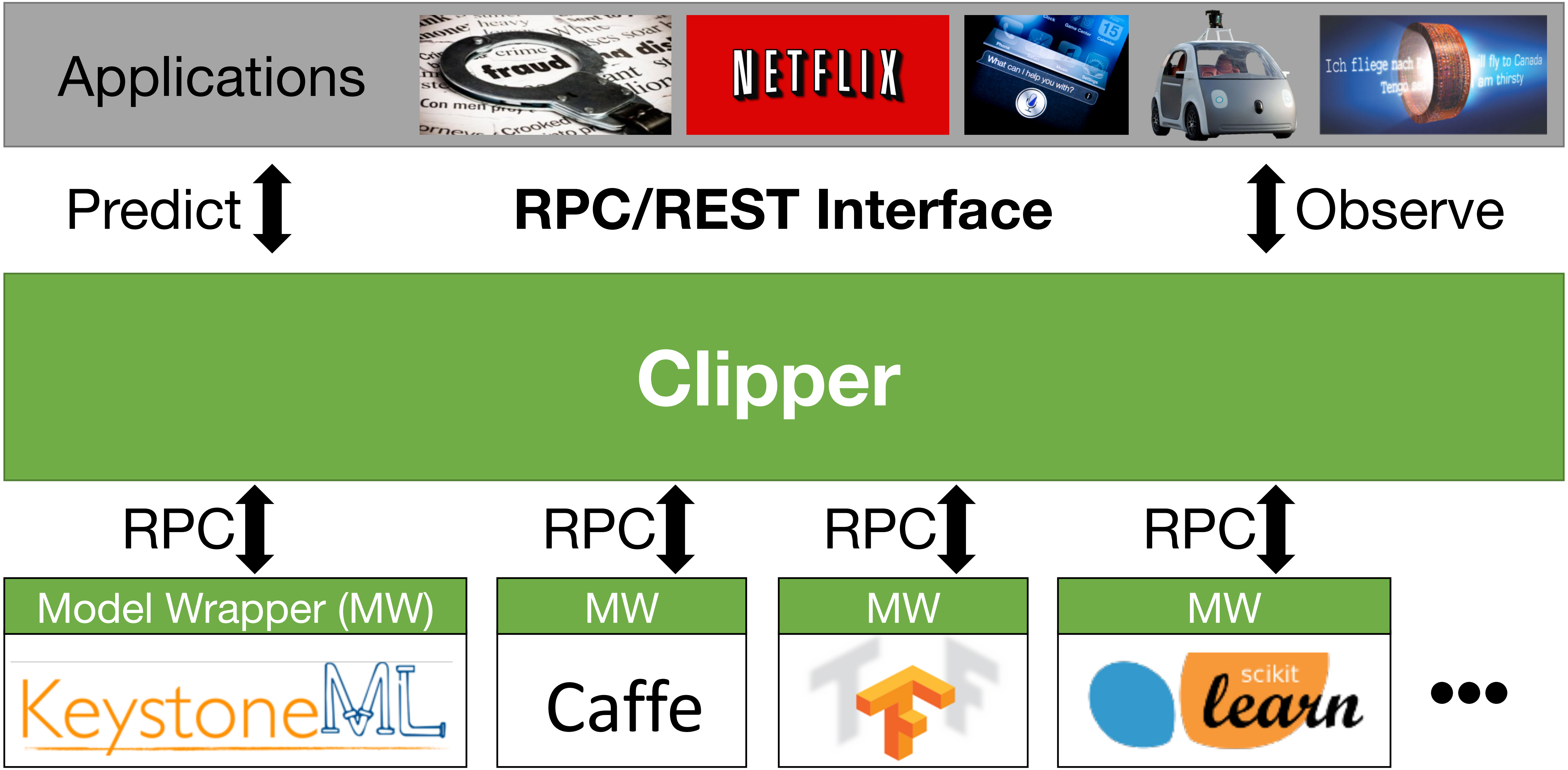
scikit learn

dmlc mxnet

VW

KALDI

Clipper Architecture



Clipper Architecture

Applications

Predict

RPC/REST Interface

Observe

Clipper

*Improve accuracy through **ensembles**, **online learning** and **personalization***

Model Selection Layer

*Provide a **common interface** to models while **bounding latency** and **maximizing throughput**.*

Model Abstraction Layer

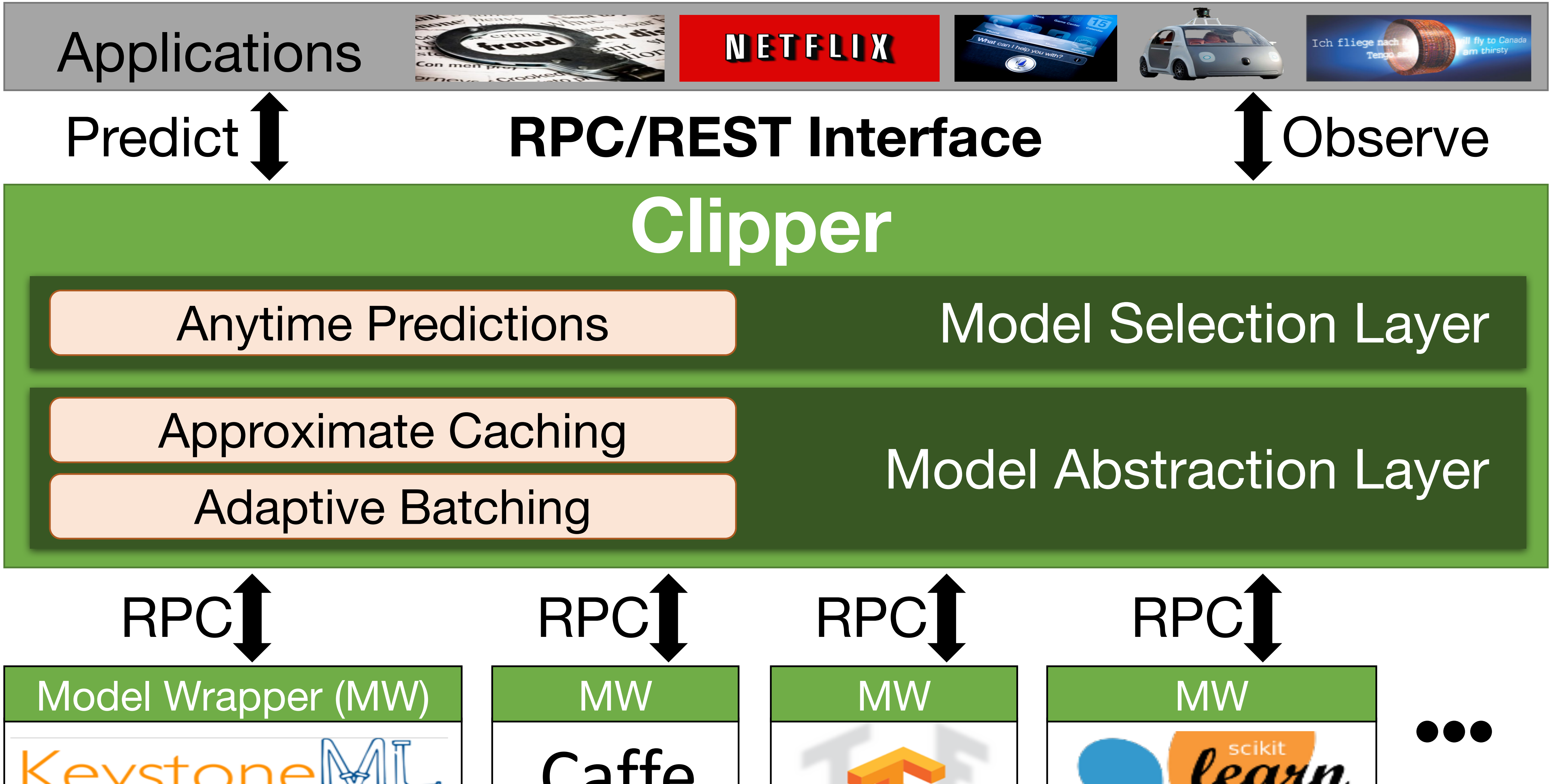
RPC RPC RPC RPC

Model Wrapper (MW) | MW | MW | MW

KeystoneML | Caffe | | scikit learn

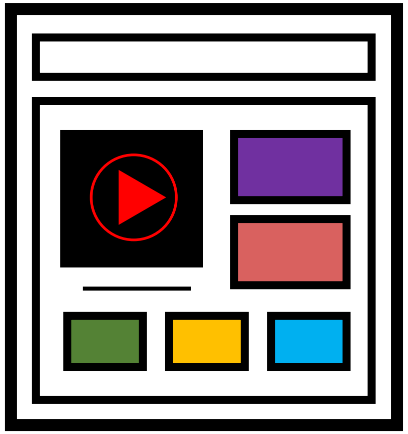
•••

Clipper Architecture



Adaptive Batching to Improve Throughput

- Why batching helps:



A single page load may generate many queries

Hardware Acceleration

GRPC

Helps amortize system overhead

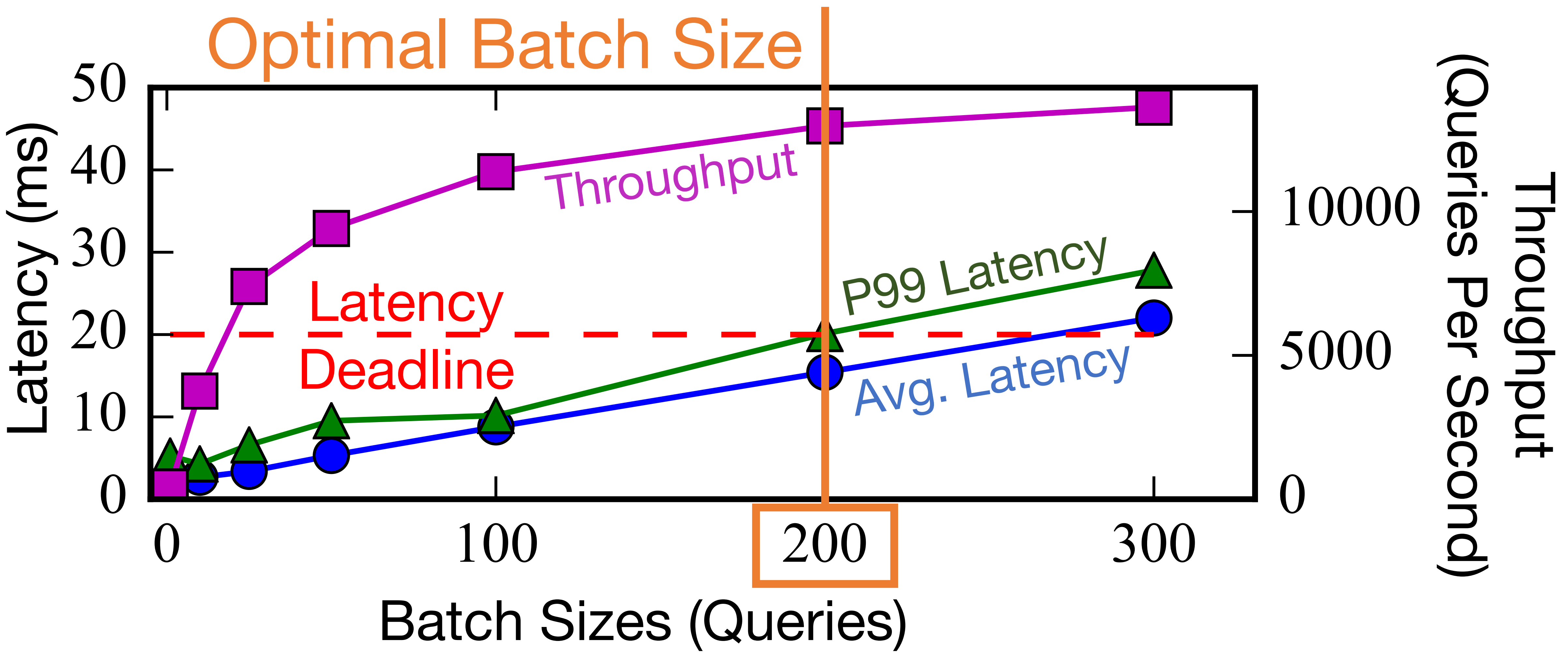
- Optimal batch depends on:
  - hardware configuration
  - model and framework
  - system load

Clipper Solution:

*be as **slow** as **allowed**…*

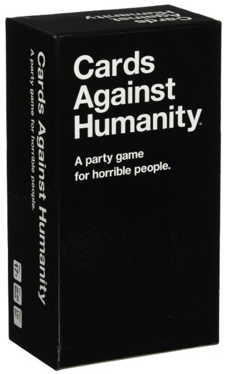
- Application specifies latency objective
- Clipper uses TCP-like tuning algorithm to **increase latency** up to the objective

Tensor Flow Conv. Net (GPU)



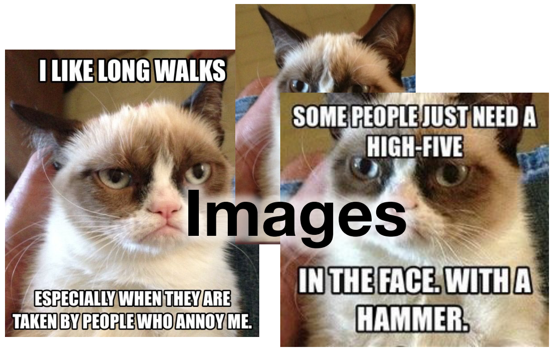
Approximate Caching to Reduce Latency

- ➢ Opportunity for caching



Popular items may be evaluated frequently

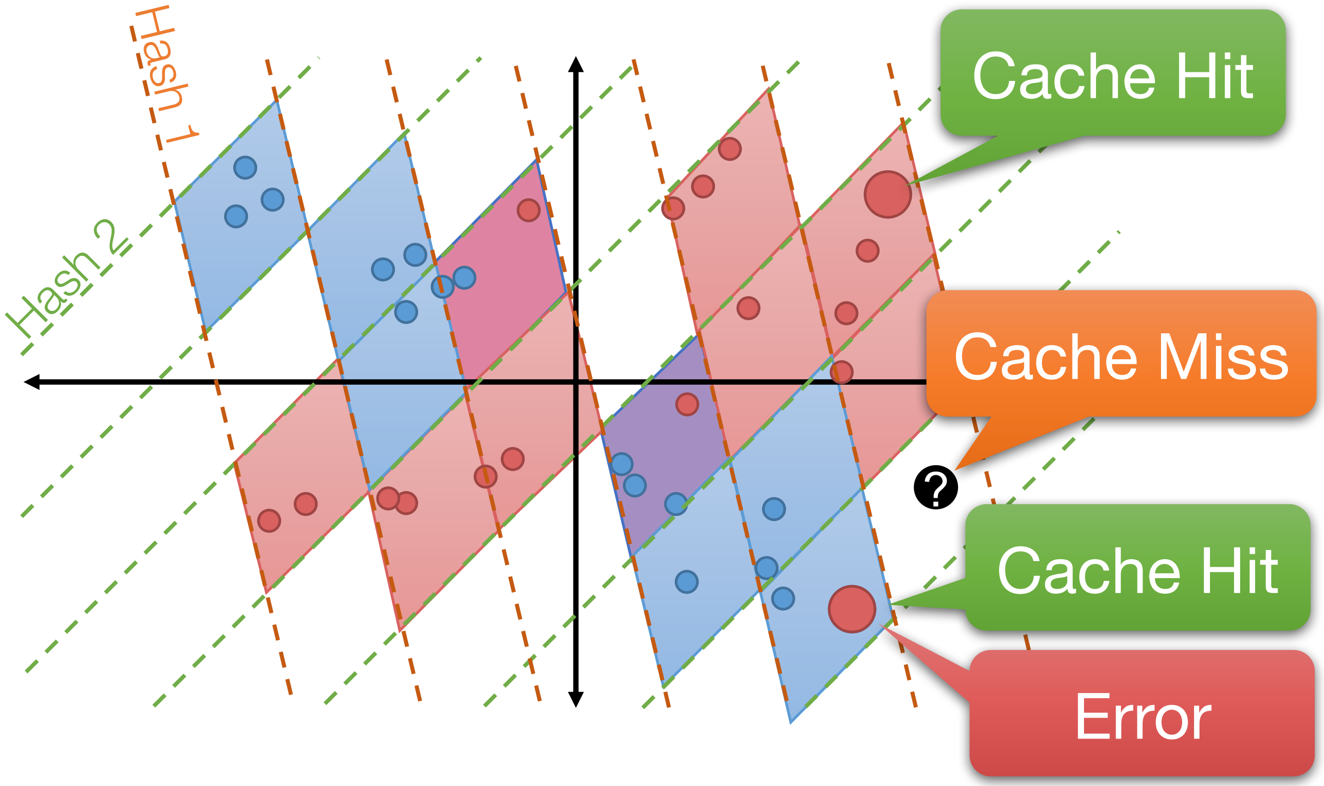
- ➢ Need for **approximation**



High Dimensional and continuous valued queries have low cache hit rate.

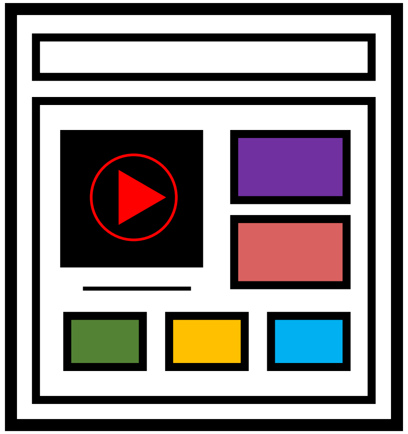
Clipper Solution: *Approximate Caching*

apply *locality sensitive hash functions*



Adaptive Batching to Improve Throughput

- Why batching helps:



A single page load may generate many queries

Hardware Acceleration

Helps amortize system overhead

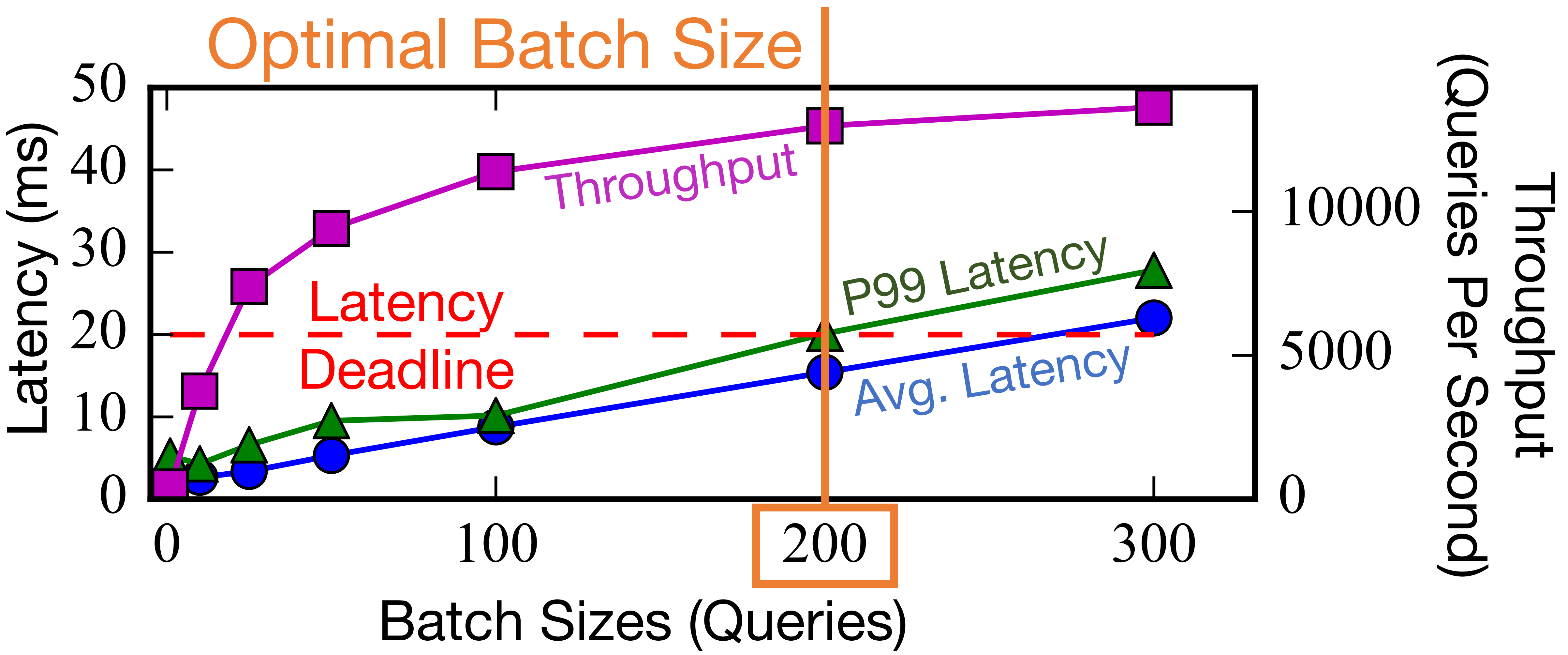
- Optimal batch depends on:
  - hardware configuration
  - model and framework
  - system load

Clipper Solution:

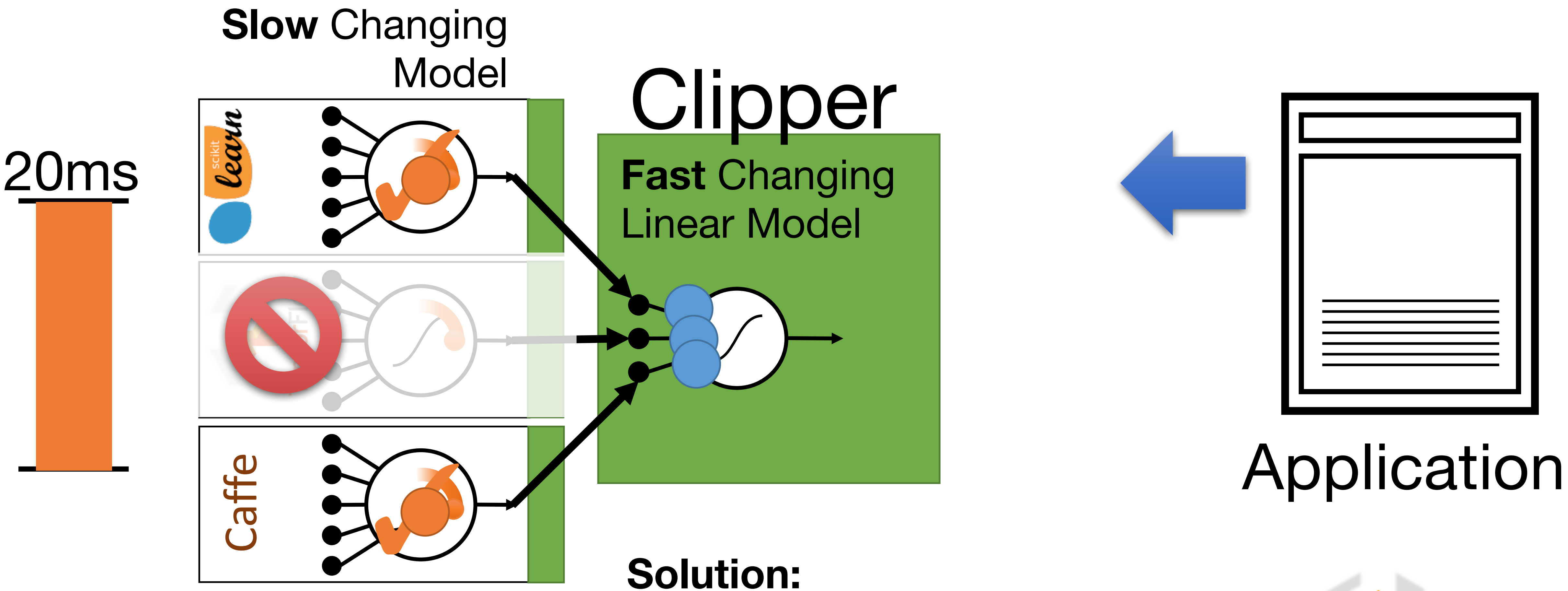
*be as **slow** as **allowed**…*

- Application specifies latency objective
- Clipper uses TCP-like tuning algorithm to **increase latency** up to the objective

Tensor Flow Conv. Net (GPU)

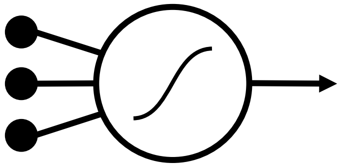


Anytime Predictions



Solution:
Replace missing prediction with an estimator ➡ E[TensorFlow (x)]

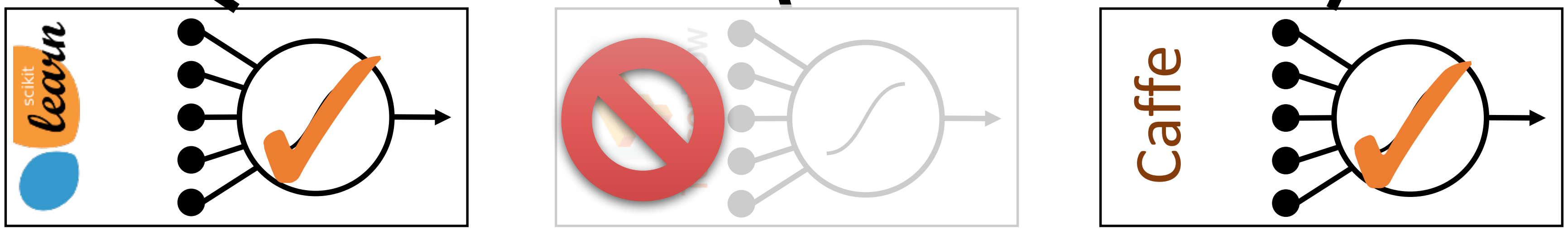
Anytime Predictions



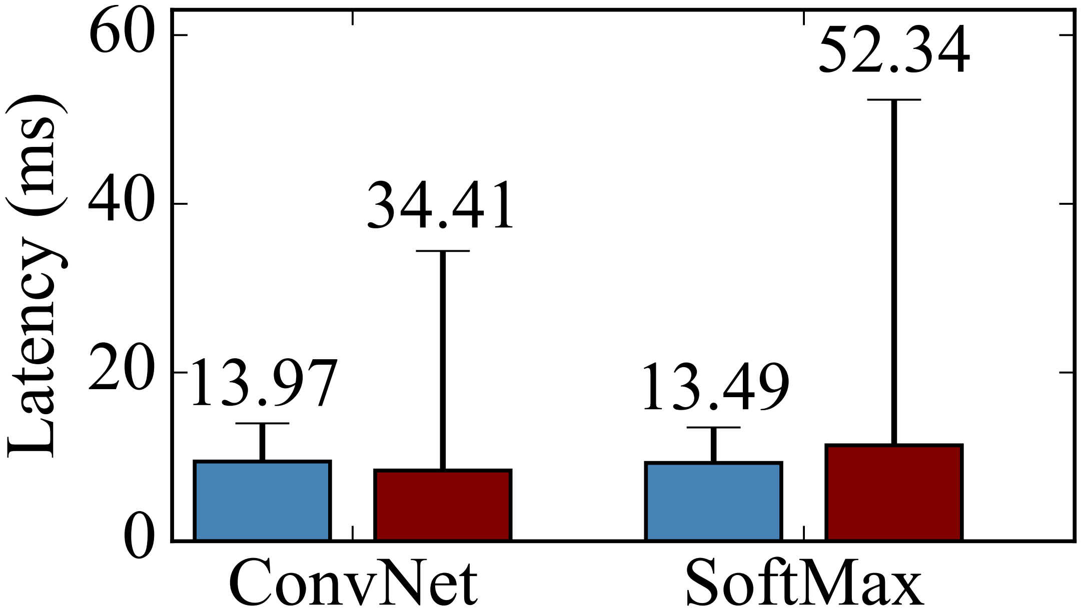
Fast Changing Model

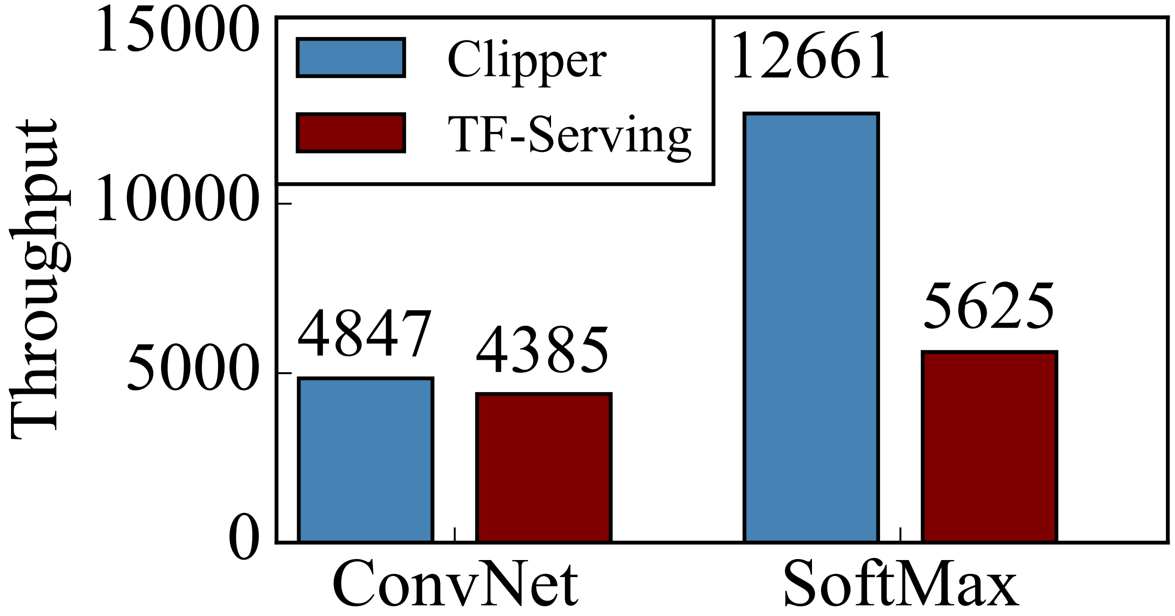
$$\mathrm{w}_{\text{scikit}}\,\underline{f_{\text{scikit}}(x)} + \mathrm{w}_{\text{TF}}\,\underline{\mathbb{E}_X\left[f_{\text{TF}}(X)\right]} + \mathrm{w}_{\text{Caffe}}\,\underline{f_{\text{Caffe}}(x)}$$

Slow Changing Model



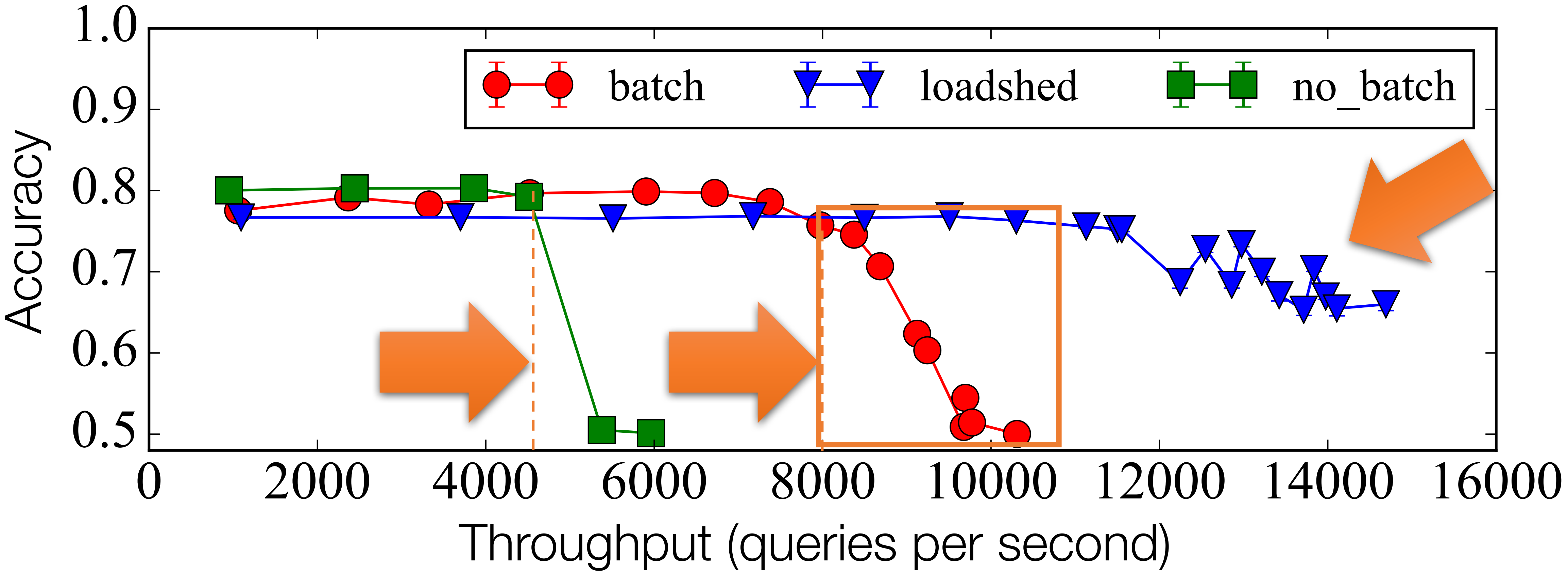
Comparison to TensorFlow Serving





Takeaway: *Clipper is able to **match the average latency** of TensorFlow Serving while reducing **tail latency (2x)** and **improving throughput (2x)***

Evaluation of Throughput Under Heavy Load



Takeaway: *Clipper is able to* ***gracefully degrade accuracy*** *to maintain availability under heavy load.*

Improved Prediction **Accuracy** (ImageNet)

System	Model	Error Rate	#Errors
Caffe	VGG	13.05%	6525
Caffe	LeNet	11.52%	5760
Caffe	ResNet	9.02%	4512
TensorFlow	Inception v3	6.18%	3088

sequence of pre-trained models

Improved Prediction **Accuracy** (ImageNet)

System			#Errors
Caffe			6525
Caffe			5760
Caffe	ResNet	9.02%	4512
TensorFlow	Inception v3	6.18%	3088
Clipper	**Ensemble**	**5.86%**	**2930**

5.2% relative improvement
in prediction accuracy!

Clipper

Clipper prediction serving system that spans multiple ML Frameworks and is designed to

- ➢ to **simplifying** model serving
- ➢ **bound latency** and **increase throughput**
- ➢ and enable **real-time learning** and **personalization**

across machine learning frameworks

Learning Systems

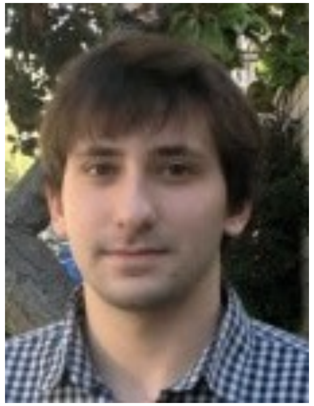
Joseph E. Gonzalez

773 Soda Hall

jegonzal@cs.berkeley.edu

Graduate students collaborators on this work:

Francois Billetti



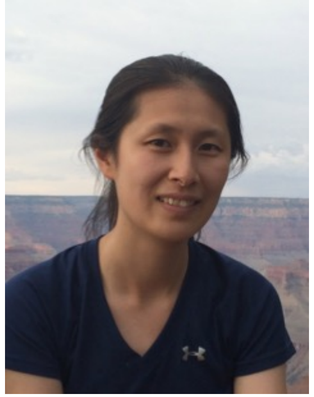
Daniel Crankshaw

Ankur Dave

Xinghao Pan

Xin Wang

Neeraja Yadwadkar



Wenting Zheng

Real-time, **I**ntelligent, and **Se**cure Systems Lab

Joseph E. Gonzalez
jegonzal@cs.berkeley.edu

Joseph Hellerstein
hellerstein@cs.berkeley.edu

Raluca Ada Popa
raluca.popa@cs.berkeley.edu

Ion Stoica
istoica@cs.berkeley.edu

RISE Lab

From live data to real-time decisions

AMP Lab

From batch data to advanced analytics

Goal

Real-time decisions

decide in ms

on live data

the current state of the environment

with strong security

privacy, confidentiality, integrity

Real-time, **I**ntelligent, and **Se**cure Systems Lab

Learn More:

- **CS294 Course** on RISE Topics
  https://ucbrise.github.io/cs294-rise-fa16/
- Early RISErs **Seminar** on **Mondays** at **9:30 AM**

Security: Protecting Models

Data is a core **asset** & models capture the **value** in data

- **Expensive**: many engineering & compute hours to develop
- Models can **reveal private information** about the data

How do we **protect models** from being stolen?

- Prevent them from being copied from devices (**DRM**? **SGX**?)
- Defend against **active learning** attacks on decision boundaries

How do we identify when models have been stolen?

- **Watermarks** in decision boundaries?