1st TVM and Deep Learning Compilation Conference

tvm

sampl

PAUL G. ALLEN SCHOOL

W

December 12, 2018

Luis Ceze

PAUL G. ALLEN SCHOOL
OF COMPUTER SCIENCE & ENGINEERING

sampl

Welcome to the 1st TVM and Deep Learning Compilation Conference!

Welcome to the 1st TVM and Deep Learning Compilation Conference!

180+ ppl!

Machine learning is amazing…

Machine learning is amazing…

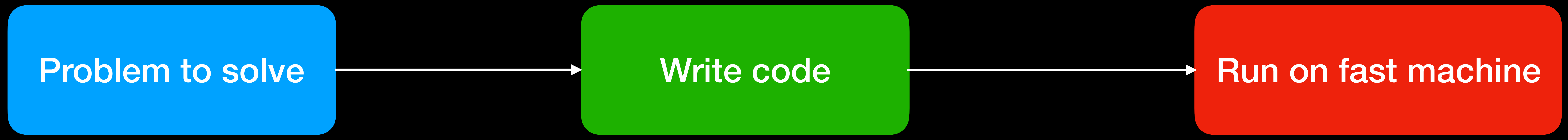
super human accuracy,
self driving cars, automated scientific discoveries…

Machine learning is amazing…

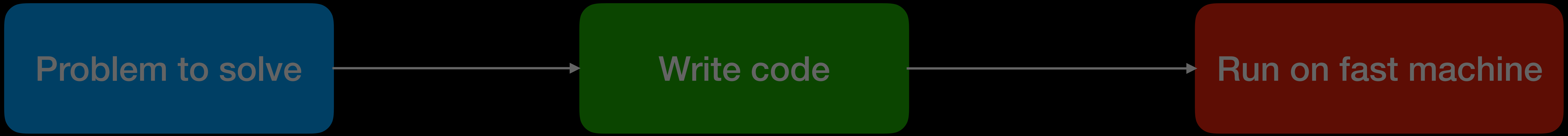
super human accuracy,
self driving cars, automated scientific discoveries…

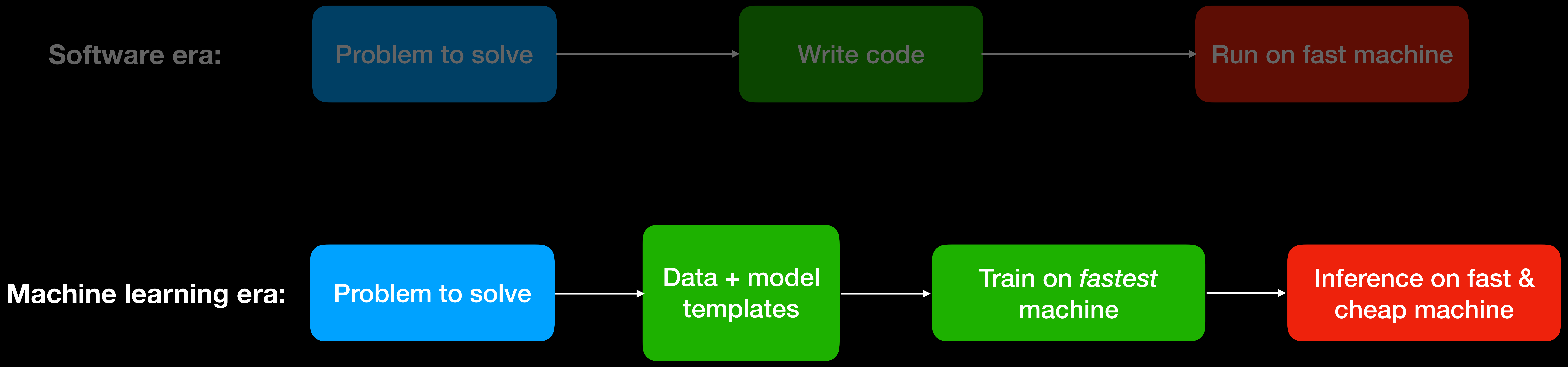
wow!

Software era:



Software era:

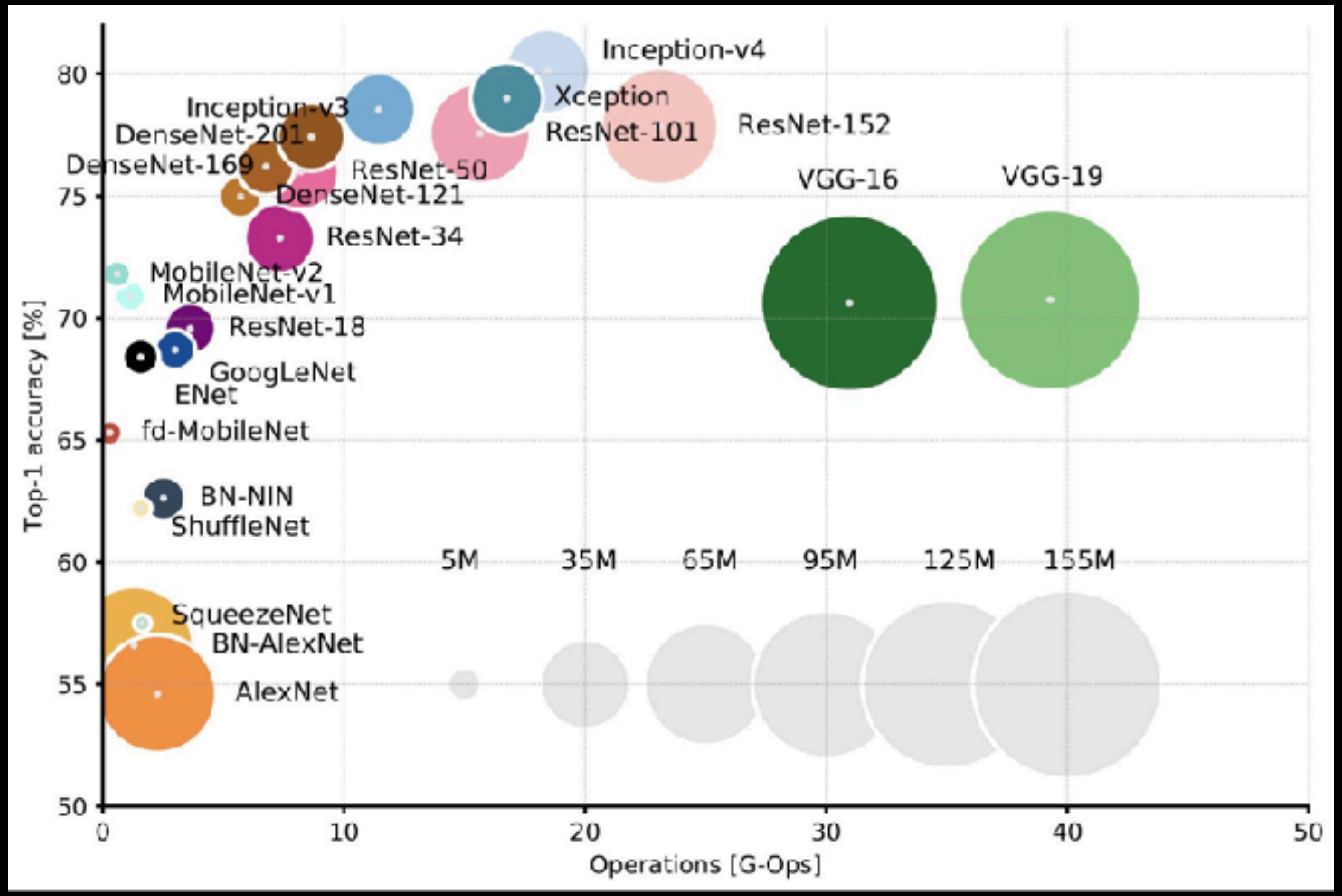



Software era:
Problem to solve
Write code
Run on fast machine
Machine learning era:
Problem to solve
Data + model templates
Train on *fastest* machine
Inference on fast & cheap machine

Software era: Problem to solve → Write code → Run on fast machine

Machine learning era: Problem to solve → Data + model templates → Train on *fastest* machine → Inference on fast & cheap machine

Model size and compute cost growing fast

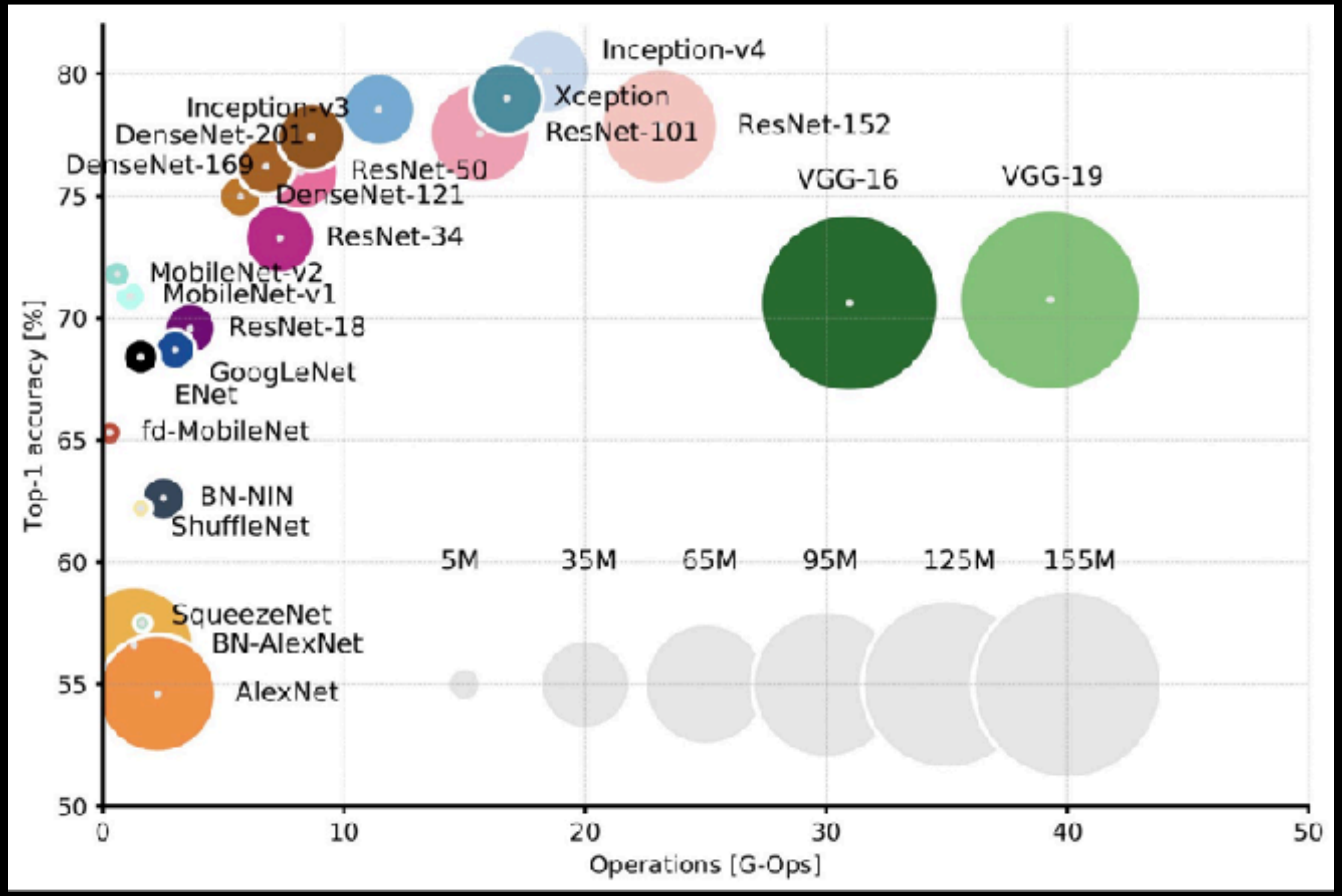


by Eugenio Culurciello

Software era: Problem to solve → Write code → Run on fast machine

Machine learning era: Problem to solve → Data + model templates → Train on *fastest* machine → Inference on fast & cheap machine

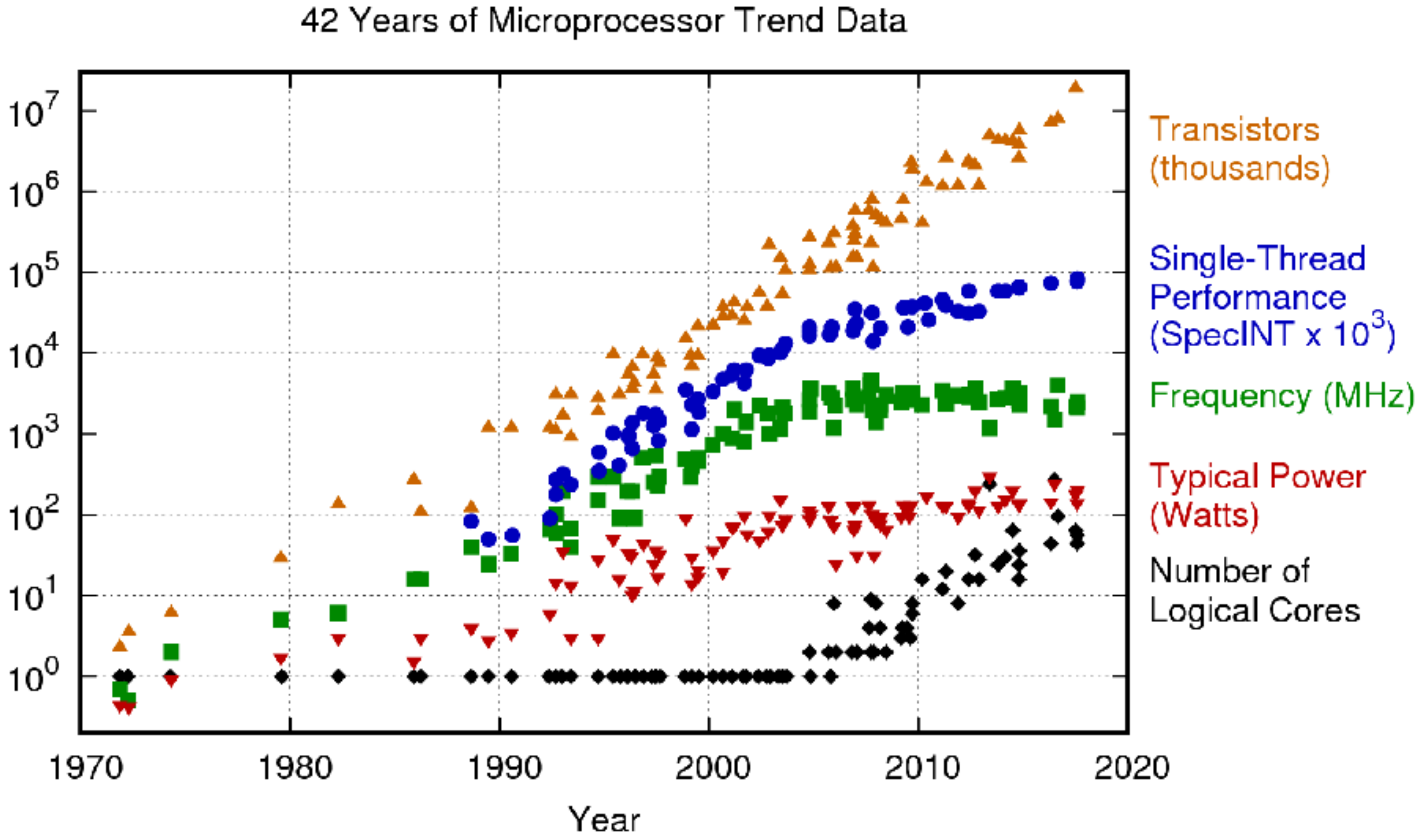
Model size and compute cost growing fast



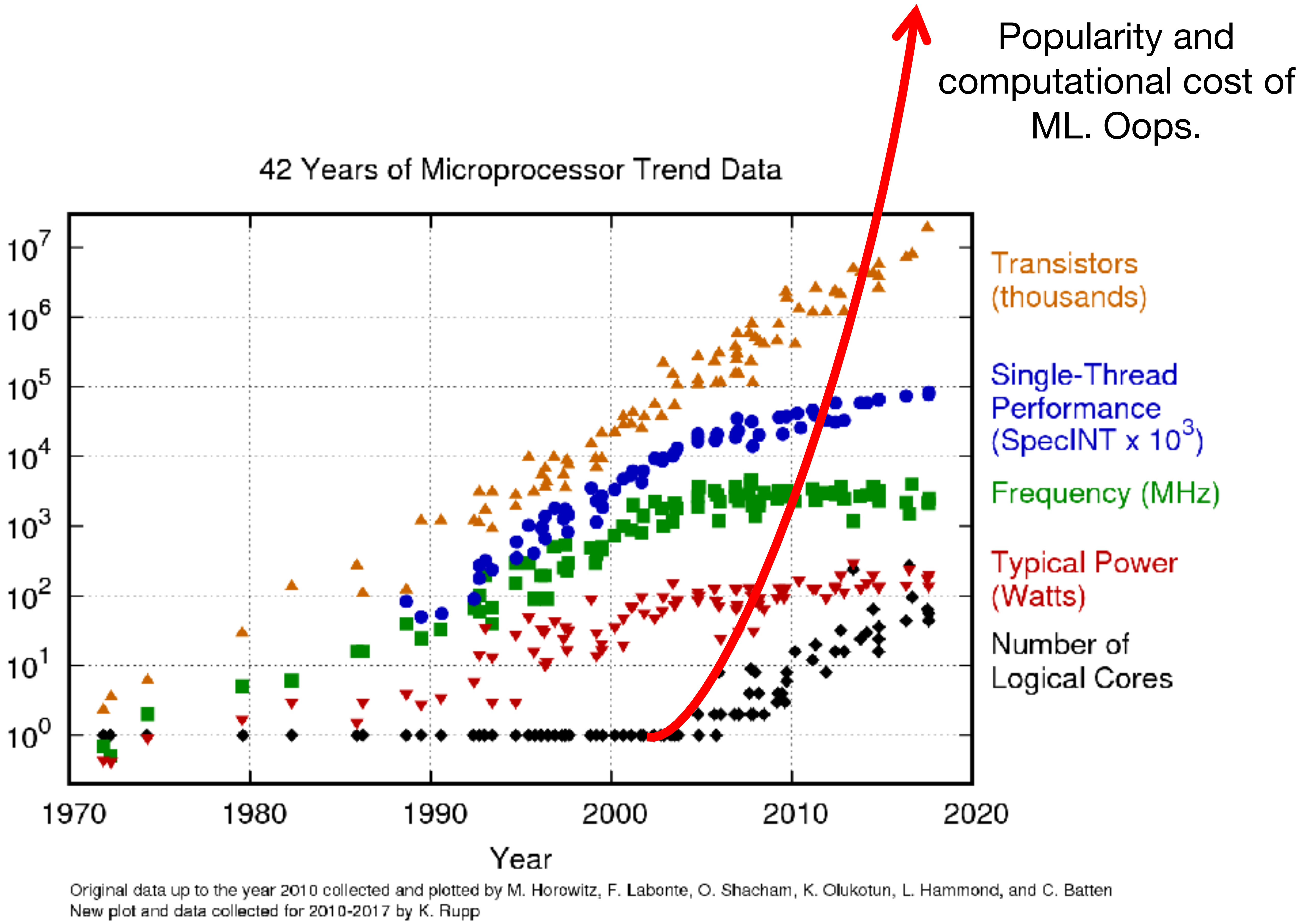
by Eugenio Culurciello

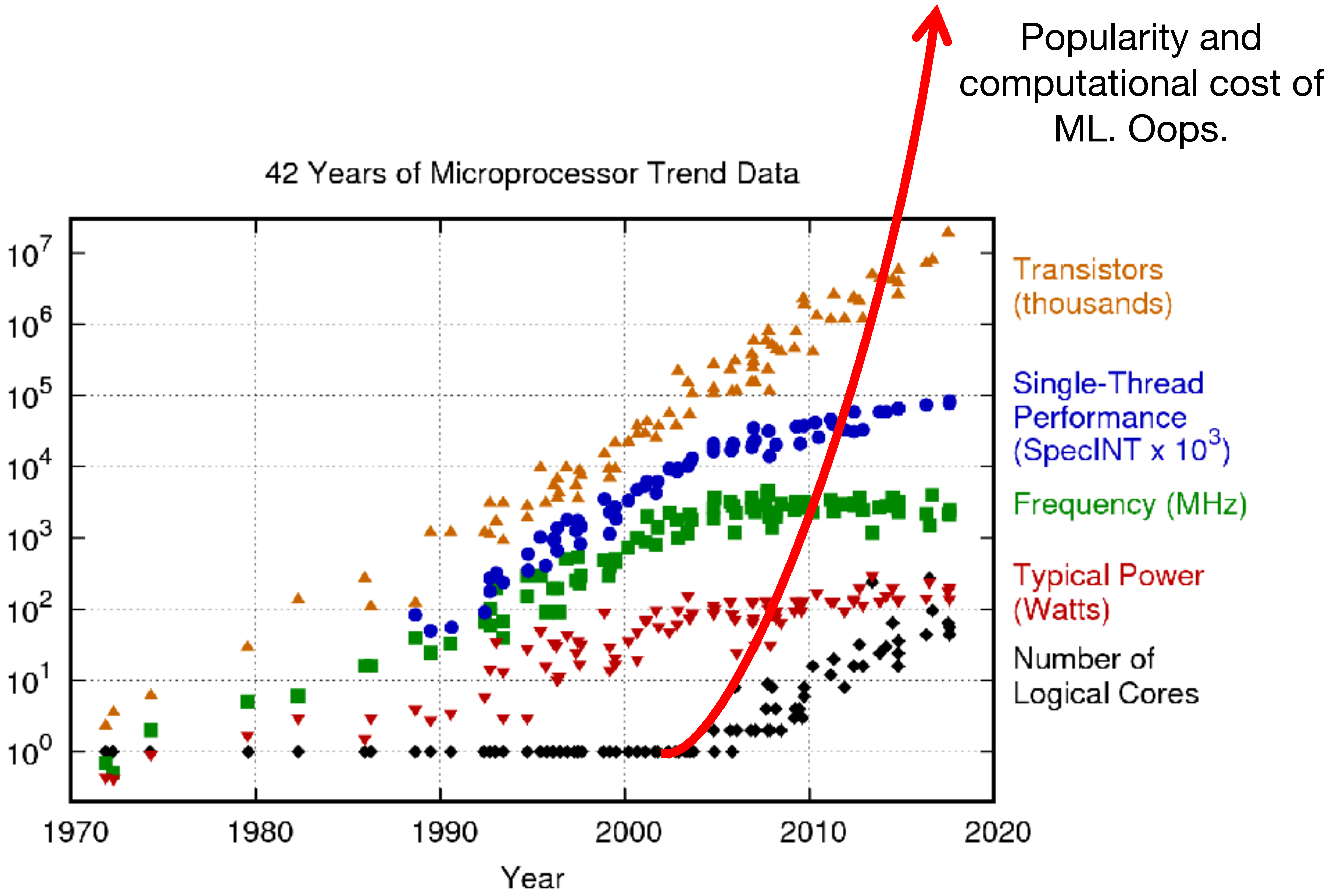
Training costs growing exponentially

by Open AI



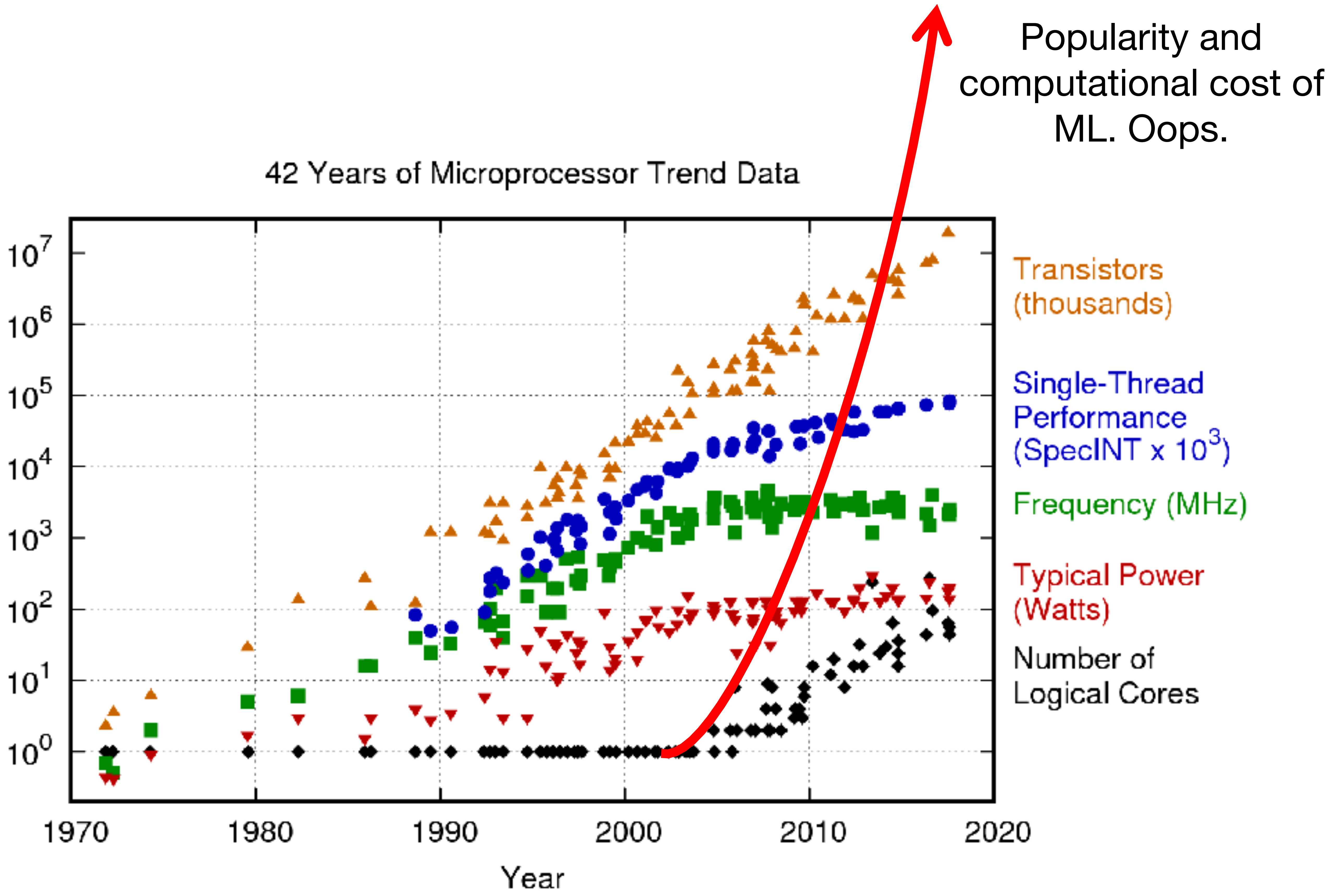
Original data up to the year 2010 collected and plotted by M. Horowitz, F. Labonte, O. Shacham, K. Olukotun, L. Hammond, and C. Batten
New plot and data collected for 2010-2017 by K. Rupp


Popularity and computational cost of ML. Oops.
42 Years of Microprocessor Trend Data
10^7
10^6
10^5
10^4
10^3
10^2
10^1
10^0
1970
1980
1990
2000
2010
2020
Year
Transistors (thousands)
Single-Thread Performance (SpecINT x 10^3)
Frequency (MHz)
Typical Power (Watts)
Number of Logical Cores
Original data up to the year 2010 collected and plotted by M. Horowitz, F. Labonte, O. Shacham, K. Olukotun, L. Hammond, and C. Batten
New plot and data collected for 2010-2017 by K. Rupp



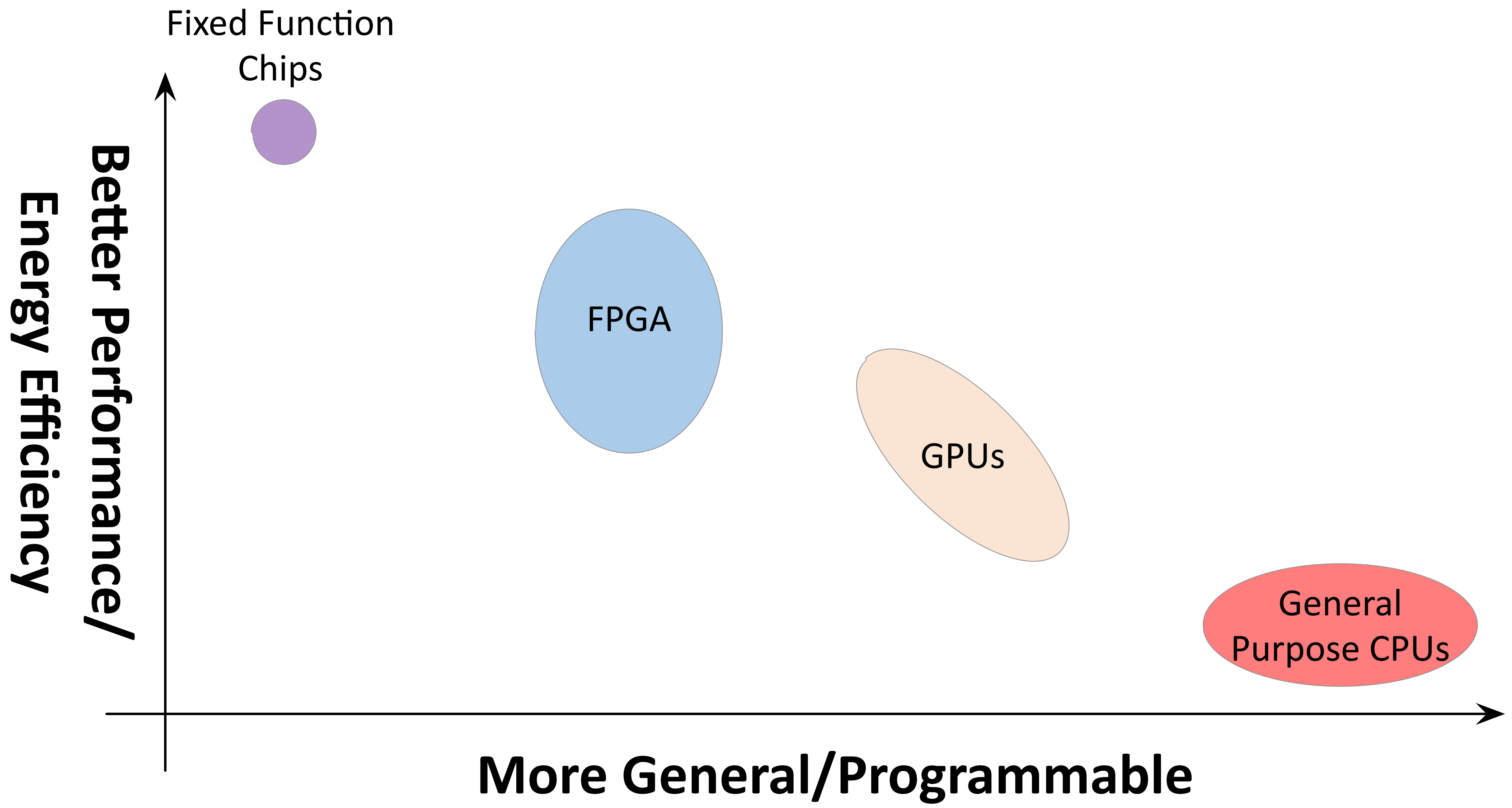
Original data up to the year 2010 collected and plotted by M. Horowitz, F. Labonte, O. Shacham, K. Olukotun, L. Hammond, and C. Batten
New plot and data collected for 2010-2017 by K. Rupp

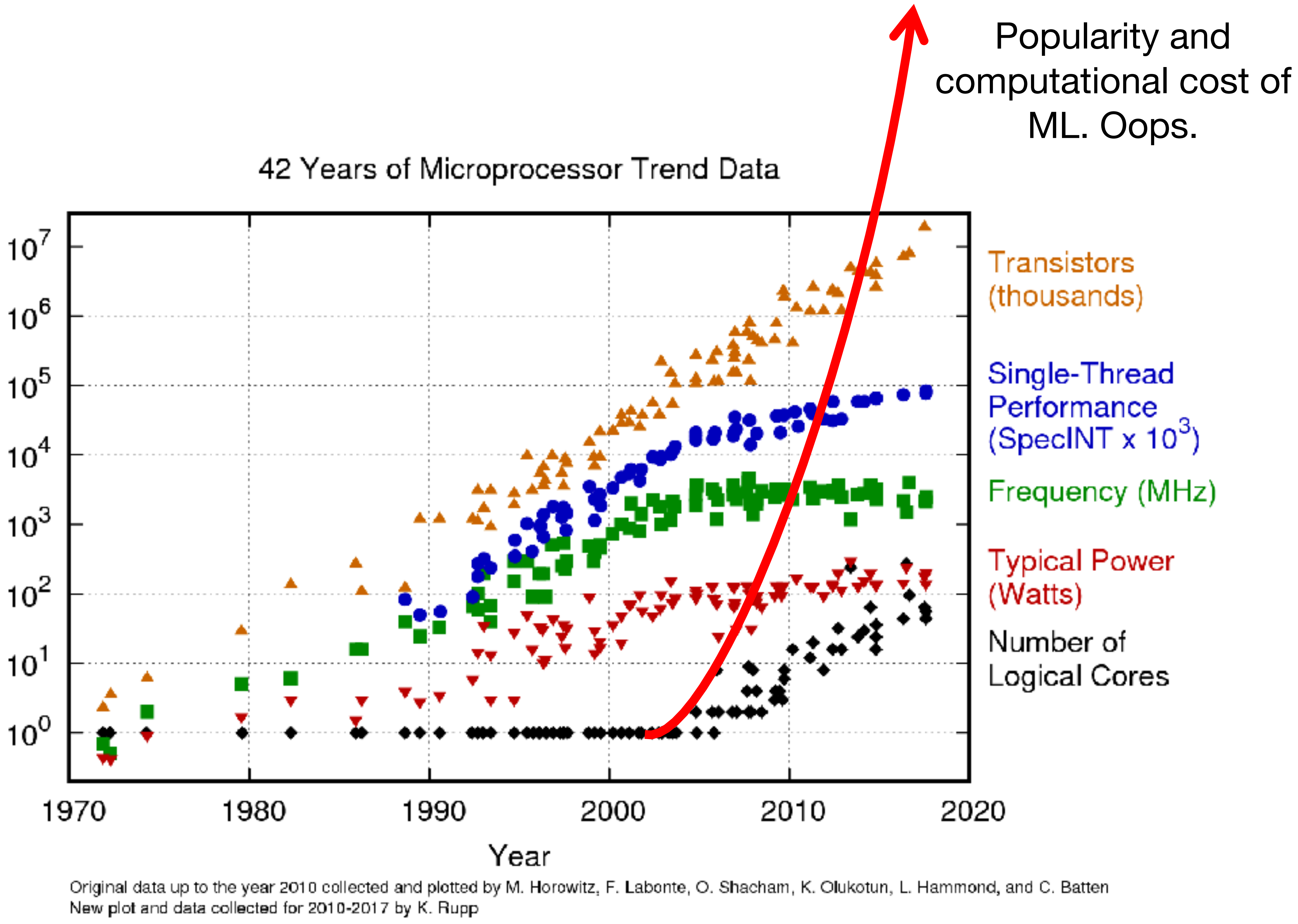
Fundamental trade-off between specialization and performance/efficiency.



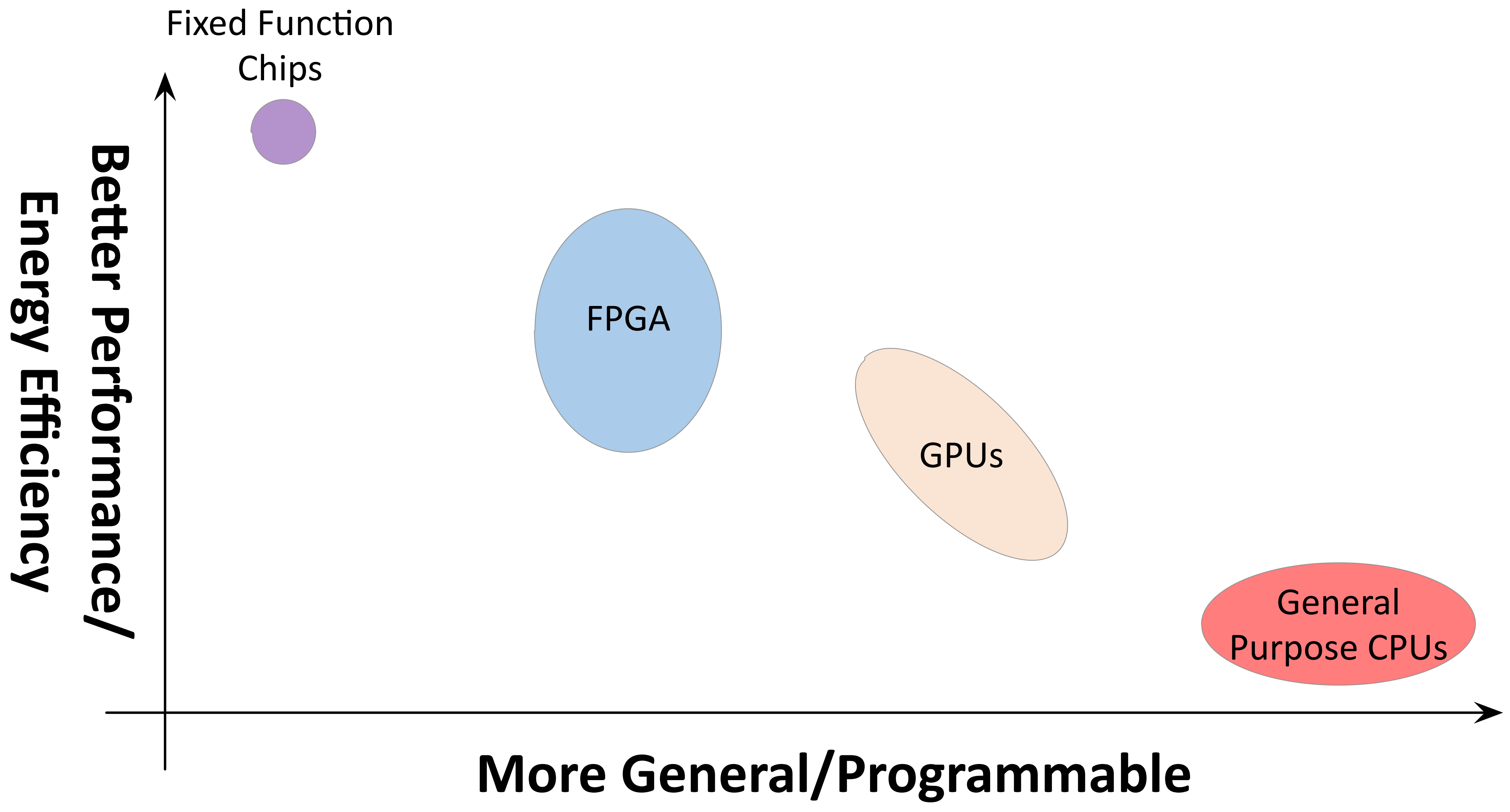
Original data up to the year 2010 collected and plotted by M. Horowitz, F. Labonte, O. Shacham, K. Olukotun, L. Hammond, and C. Batten
New plot and data collected for 2010-2017 by K. Rupp

Fundamental trade-off between specialization and performance/efficiency.

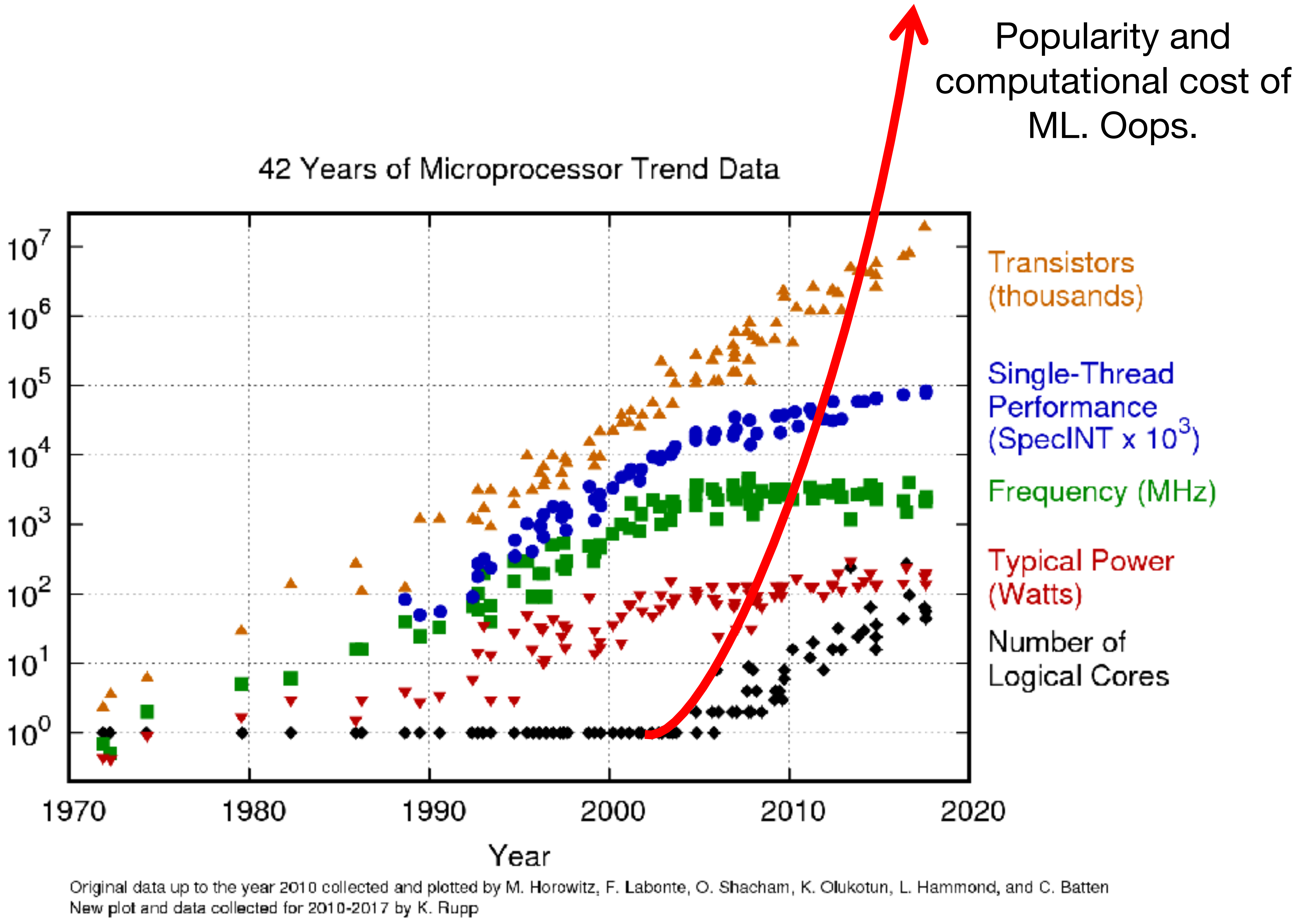




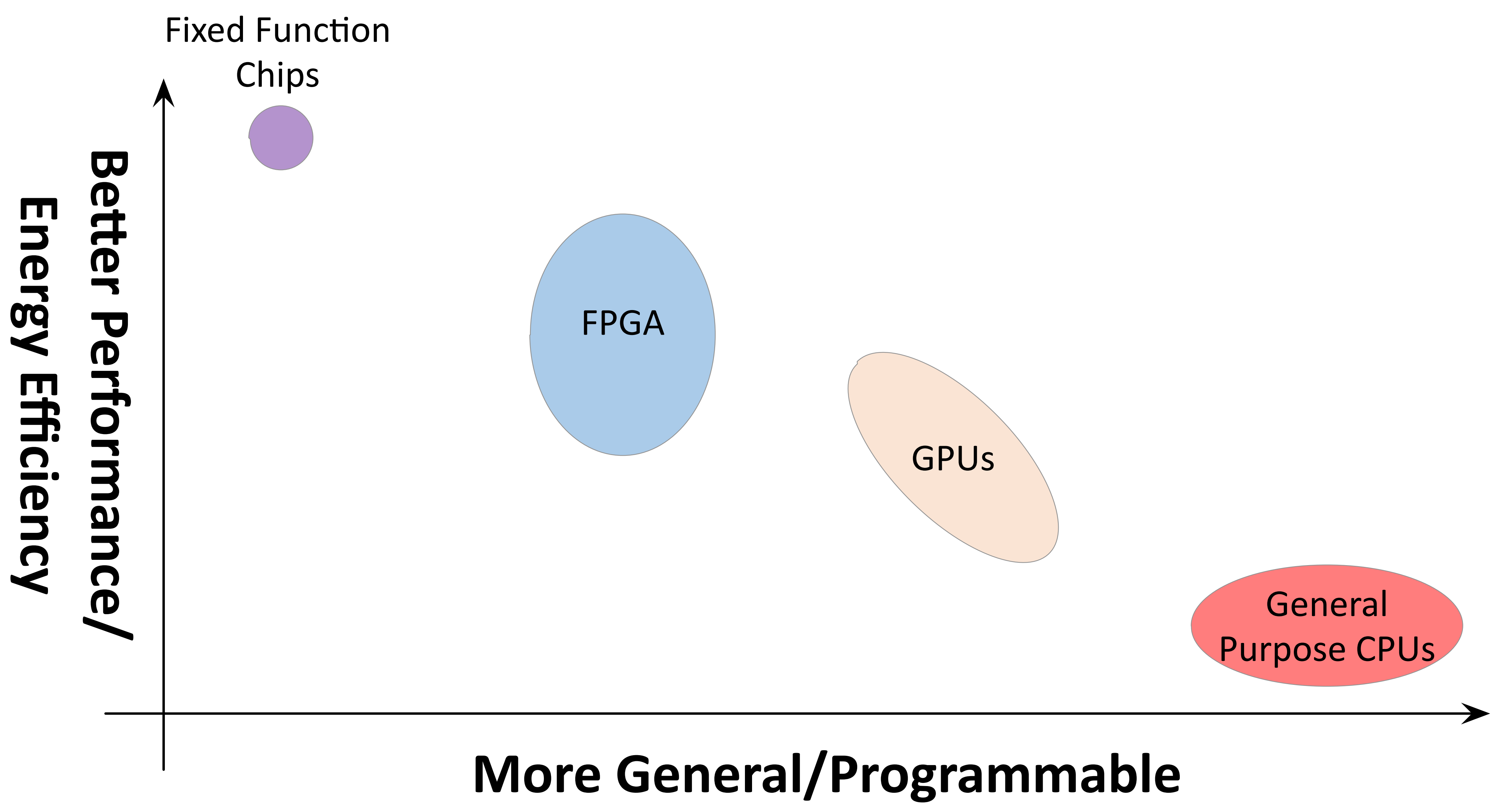
Fundamental trade-off between specialization and performance/efficiency.



Machine learning algorithms are **relatively** simple to implement in HW… great!



Fundamental trade-off between specialization and performance/efficiency.



Machine learning algorithms are **relatively** simple to implement in HW… great!

Machine Learning Makes Computer Architecture Cool Again!


A11
intel
Xeon
processor
...

+~50 startups


ZYNQ

+~50 startups

Models:

CNN

GAN

RNN

MLP

DQNN

...

intel NVIDIA XILINX Microsoft QUALCOMM amazon Google HUAWEI

+~50 startups

Models: CNN GAN RNN MLP DQNN

Frameworks:

+~50 startups

Models: CNN GAN RNN MLP DQNN

Frameworks:

Challenge: Efficiently deploying deep learning everywhere

...

intel NVIDIA XILINX Microsoft QUALCOMM amazon Google HUAWEI

+~50 startups

Gaurav Kapoor, Core Machine Learning

HW+SW optimization is key for efficiency

HW+SW optimization is key for efficiency

Lots of hand-tuning, full automation would be a holy grail

W sampl

Academic group focused on **S**ystems + Computer **A**rchitecture + **M**achine Learning + **P**rogramming Languages


Luis Ceze
Professor


Carlos Guestrin
Professor


Arvind Krishnamurthy
Professor


Zachary Tatlock
Assistant Professor


Meghan Cowan


Thierry Moreau


Eddie Yan


Luis Vega


Jared Roesch


Steven Lyubomirsky


Tianqi Chen


Haichen Shen


Liang Luo


Logan Weber


Pratyush Patel


Josh Fromm


Gus Smith


Ziheng Jiang


Josh Pollock


Marisa Kirisame


Lianmin Zheng


Seungyeop Han


Jacob Nelson


Amar Phanishayee

Academic group focused on **S**ystems + Computer **A**rchitecture + **M**achine Learning + **P**rogramming Languages

Luis Ceze
Professor

Carlos Guestrin
Professor

Arvind Krishnamurthy
Professor

Zachary Tatlock
Assistant Professor

Meghan Cowan

Thierry Moreau

Eddie Yan

Luis Vega

Jared Roesch

Steven Lyubomirsky

Tianqi Chen

Haichen Shen

Liang Luo

Logan Weber

Pratyush Patel

Josh Fromm

Gus Smith

Ziheng Jiang

Josh Pollock

Marisa Kirisame

Lianmin Zheng

Seungyeop Han

Jacob Nelson

Amar Phanishayee

W sampl

Academic group focused on **S**ystems + Computer **A**rchitecture + **M**achine Learning + **P**rogramming **L**anguages

Luis Ceze
Professor

Carlos Guestrin
Professor

Arvind Krishnamurthy
Professor

Zachary Tatlock
Assistant Professor

Meghan Cowan

Thierry Moreau

Eddie Yan

Luis Vega

Jared Roesch

Steven Lyubomirsky

Tianqi Chen

Haichen Shen

Liang Luo

Logan Weber

Pratyush Patel

Josh Fromm

Gus Smith

Ziheng Jiang

Josh Pollock

Marisa Kirisame

Lianmin Zheng

Seungyeop Han

Jacob Nelson

Amar Phanishayee

Cornell University · UCLA · Berkeley University of California · The University of Texas at Austin

intel · Apple · HUAWEI · amazon · Microsoft · XILINX

PL: High-level support for future ML applications

Compilers: Extensible support for future models, optimizations and hardware architectures

Systems: On-device and cloud-based training, distributed systems for ML

Computer Architecture: Extensible, energy efficient hardware designs for inference and training

ML for Systems: Automatic Learning-Based Design and Optimizations

PL: High-level support for future ML applications

Compilers: Extensible support for future models, optimizations and hardware architectures

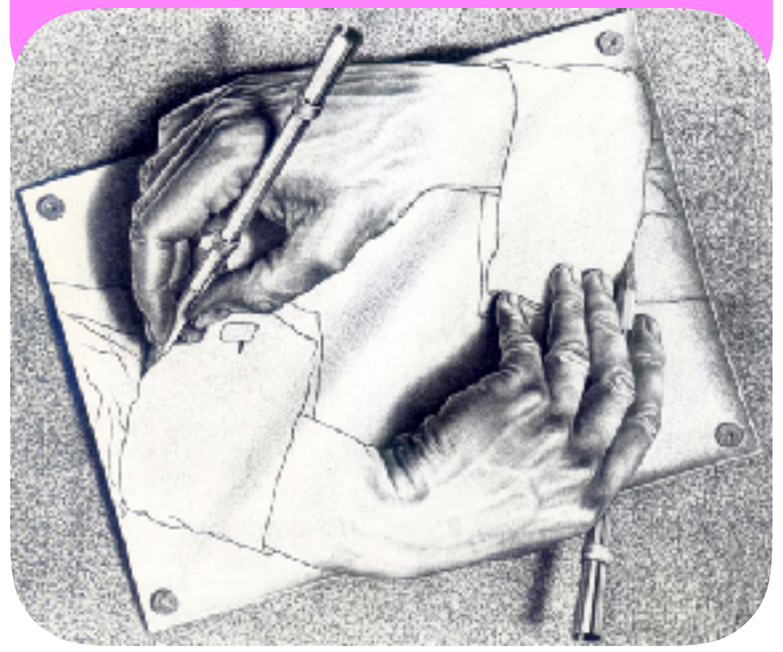
Systems: On-device and cloud-based training, distributed systems for ML

Computer Architecture: Extensible, energy efficient hardware designs for inference and training

sampl

ML for Systems: Automatic Learning-Based Design and Optimizations

ML for better ML systems!



PL: High-level support for future ML applications

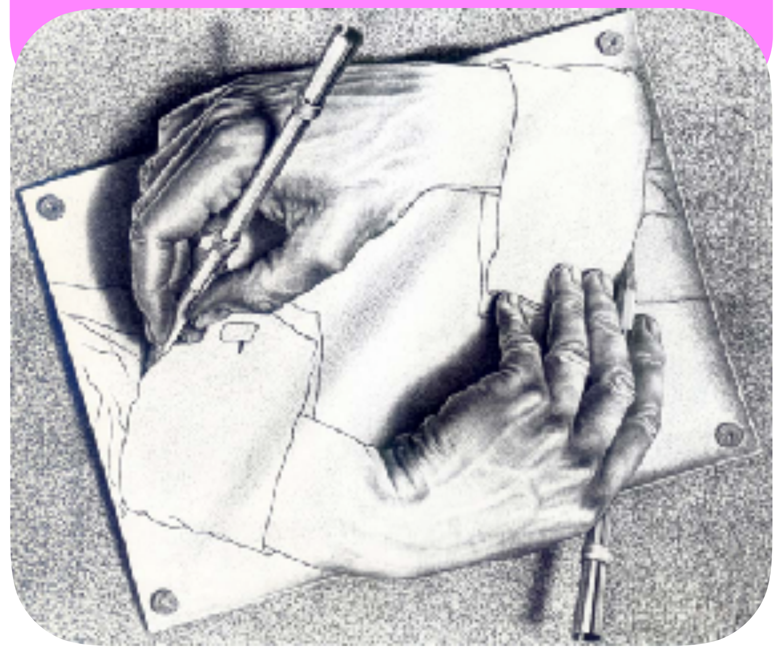
Compilers: Extensible support for future models, optimizations and hardware architectures

Systems: On-device and cloud-based training, distributed systems for ML

Computer Architecture: Extensible, energy efficient hardware designs for inference and training

ML for Systems: Automatic Learning-Based Design and Optimizations

ML for better ML systems!



PL: High-level support for future ML applications

Compilers: Extensible support for future models, optimizations and hardware architectures

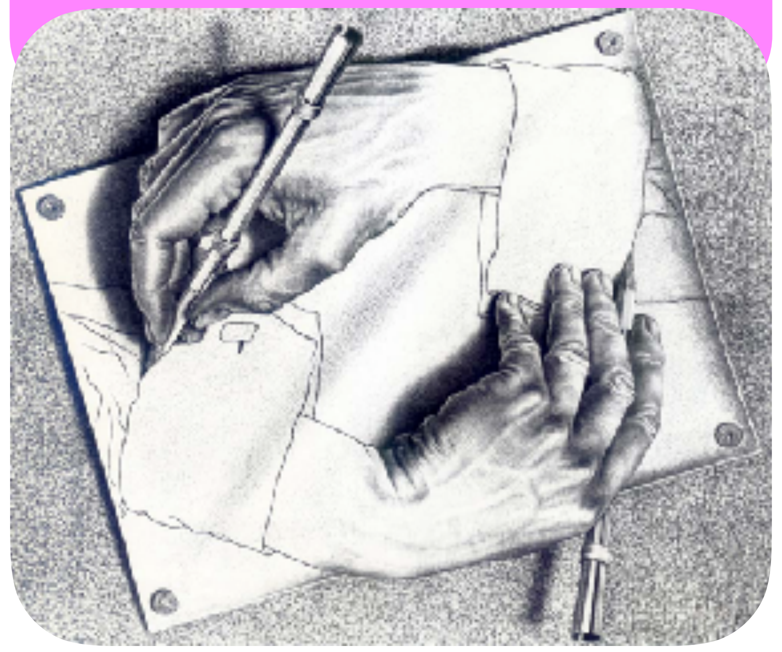
Systems: On-device and cloud-based training, distributed systems for ML

Computer Architecture: Extensible, energy efficient hardware designs for inference and training

sampl

ML for Systems: Automatic Learning-Based Design and Optimizations

ML for better ML systems!



PL: High-level support for future ML applications

Compilers: Extensible support for future models, optimizations and hardware architectures

Systems: On-device and cloud-based training, distributed systems for ML

Computer Architecture: Extensible, energy efficient hardware designs for inference and training

Open source when ready

Provides infrastructure

Open Source Deployment

Open source compilers have transformed our industry

First major open source
compiler collection

Open source compilers have transformed our industry

First major open source compiler collection

LLVM: Higher-Level IR, new optimizations, easier extensibility

Open source compilers have transformed our industry

In the age of domain-specialized systems…

First major open source compiler collection

LLVM: Higher-Level IR, new optimizations, easier extensibility

Open source compilers have transformed our industry

In the age of domain-specialized systems…

Specialized compiler stack
for Deep Learning

tvm

GCC

First major open source
compiler collection

LLVM: Higher-Level IR, new
optimizations, easier extensibility

End the tyranny of closed deep learning systems!

Tianqi Chen

PAUL G. ALLEN SCHOOL
OF COMPUTER SCIENCE & ENGINEERING

sampl

tvm

High-Level Differentiable IR

High-Level Differentiable IR

Tensor Expression IR


High-Level Differentiable IR
Tensor Expression IR
LLVM, CUDA, Metal

tvm

High-Level Differentiable IR

Tensor Expression IR

LLVM, CUDA, Metal

VTA

Edge FPGA

Cloud FPGA

ASIC

```python
import tvm
from tvm import relay

graph, params =
   frontend.from_keras(keras_resnet50)
graph, lib, params =
     relay.build(graph, target)
```

Compile

```python
import tvm
from tvm import relay

graph, params =
   frontend.from_keras(keras_resnet50)
graph, lib, params =
     relay.build(graph, target)
```

Compile

tvm

Deploy

```
module = runtime.create(graph, lib, tvm.gpu(0))
module.set_input(**params)
module.run(data=data_array)
output = tvm.nd.empty(out_shape, ctx=tvm.gpu(0))
module.get_output(0, output)
```

input

Deployable Module

prediction **tabby, tabby cat**

```
import tvm
from tvm import relay

graph, params =
   frontend.from_keras(keras_resnet50)
graph, lib, params =
     relay.build(graph, target)
```

Compile

JS Java python

tvm

```python
import tvm
from tvm import relay

graph, params =
   frontend.from_keras(keras_resnet50)
graph, lib, params =
     relay.build(graph, target)
```

Compile

Deploy

```python
module = runtime.create(graph, lib, tvm.gpu(0))
module.set_input(**params)
module.run(data=data_array)
output = tvm.nd.empty(out_shape, ctx=tvm.gpu(0))
module.get_output(0, output)
```

input

Deployable Module

prediction **tabby, tabby cat**

On languages and platforms you choose

JS Java python iOS

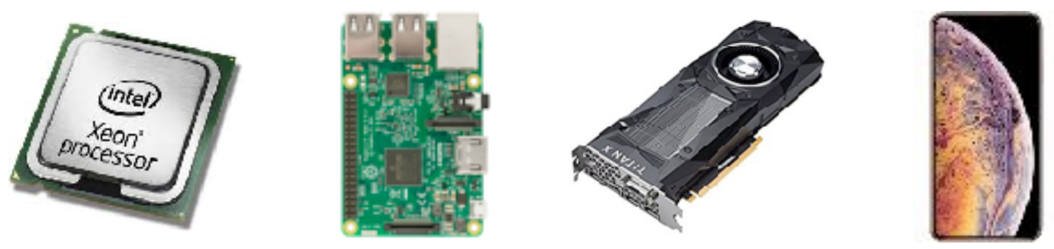
tvm Automated by Machine Learning

High-Level Differentiable IR

Tensor Expression IR

LLVM, CUDA, Metal

VTA

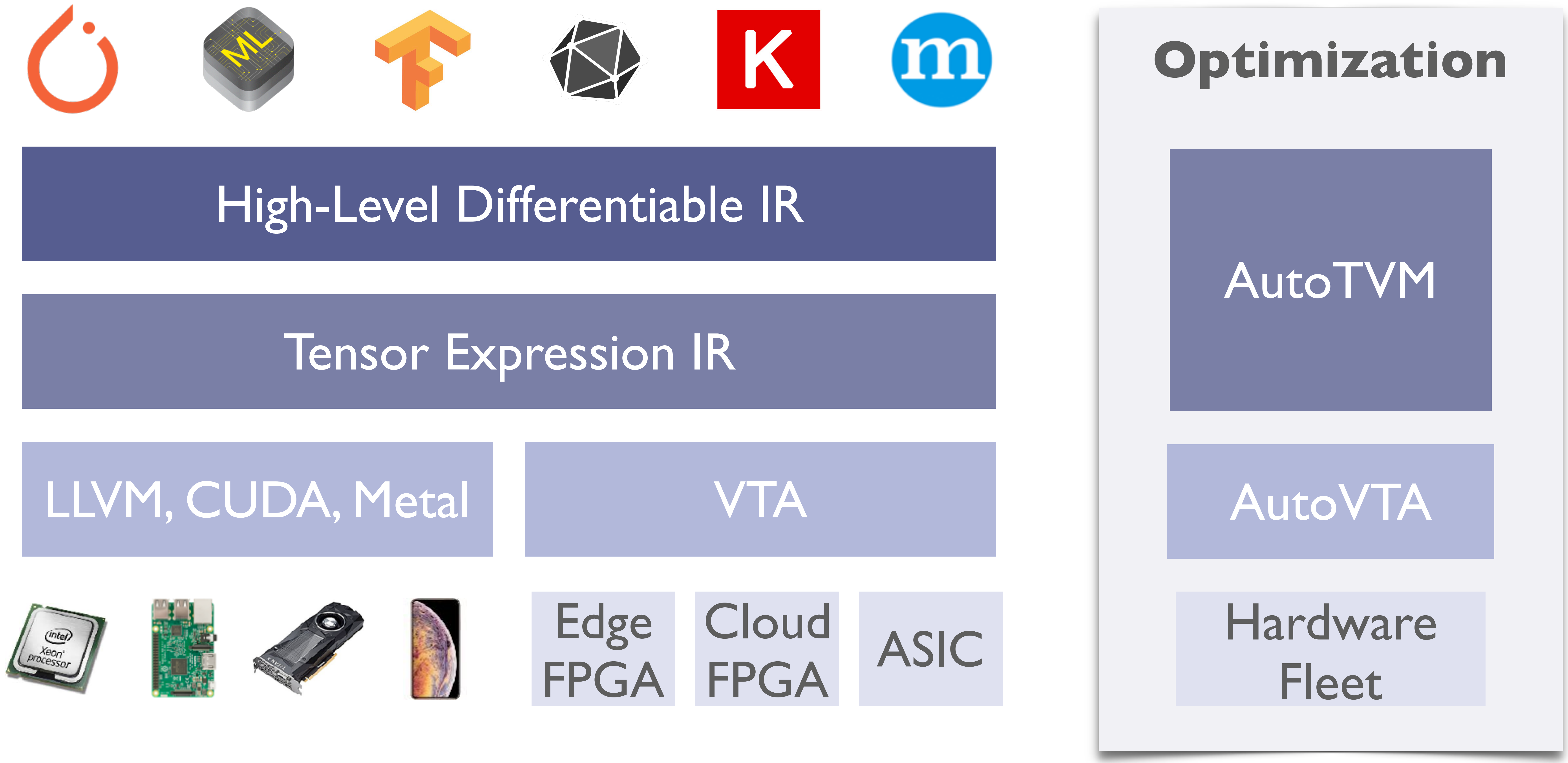


Edge FPGA

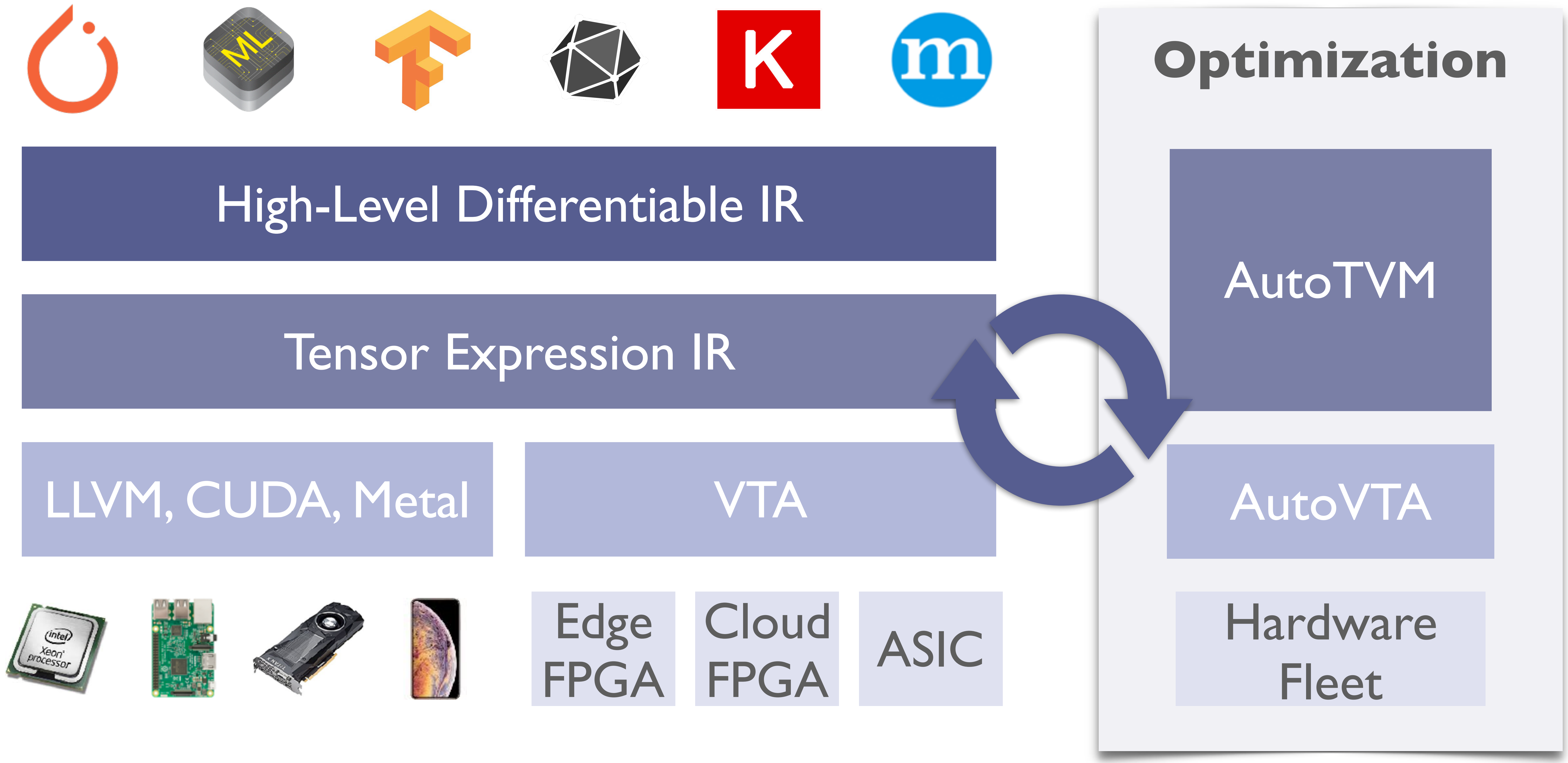
Cloud FPGA

ASIC

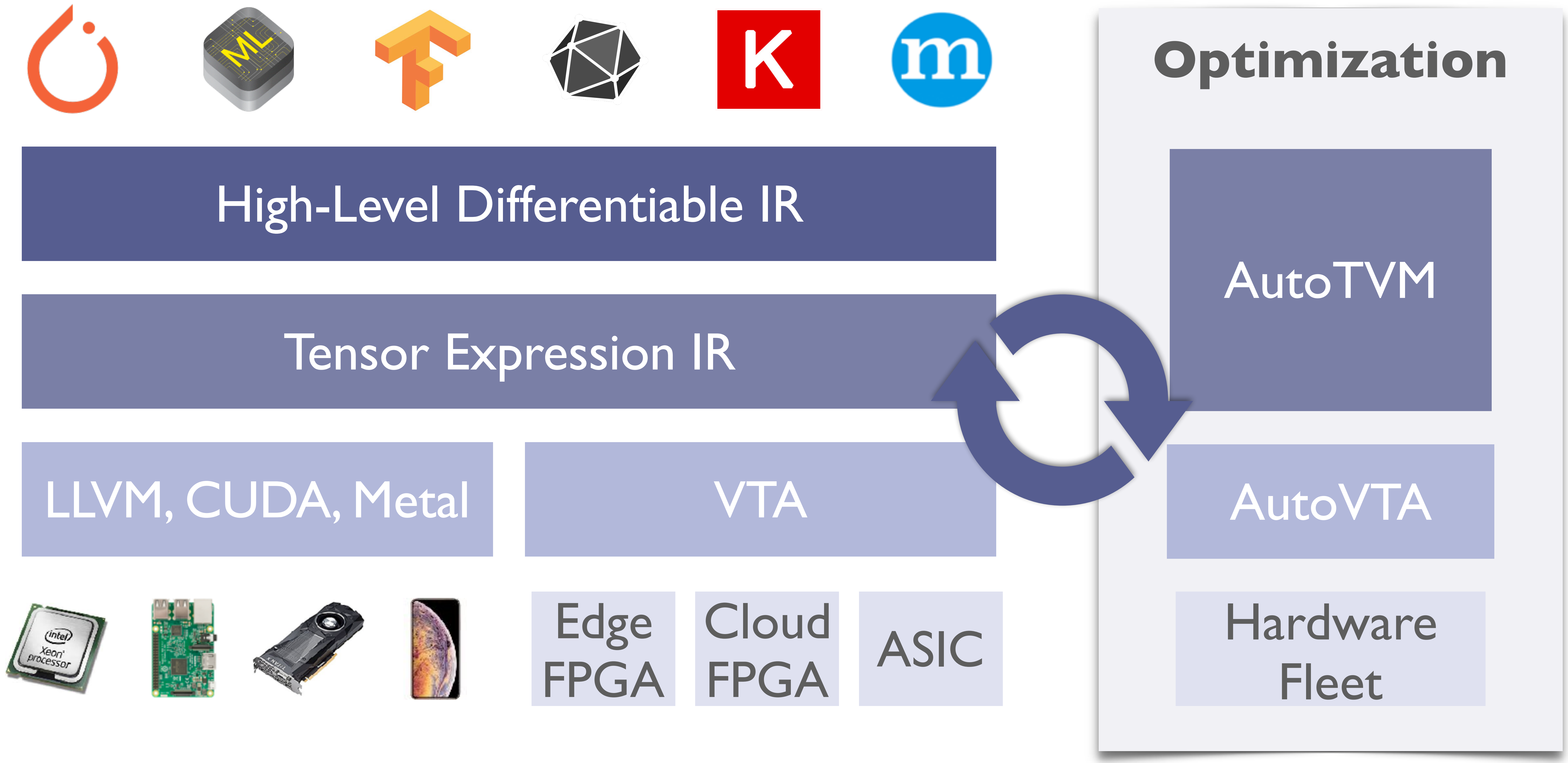
tvm Automated by Machine Learning



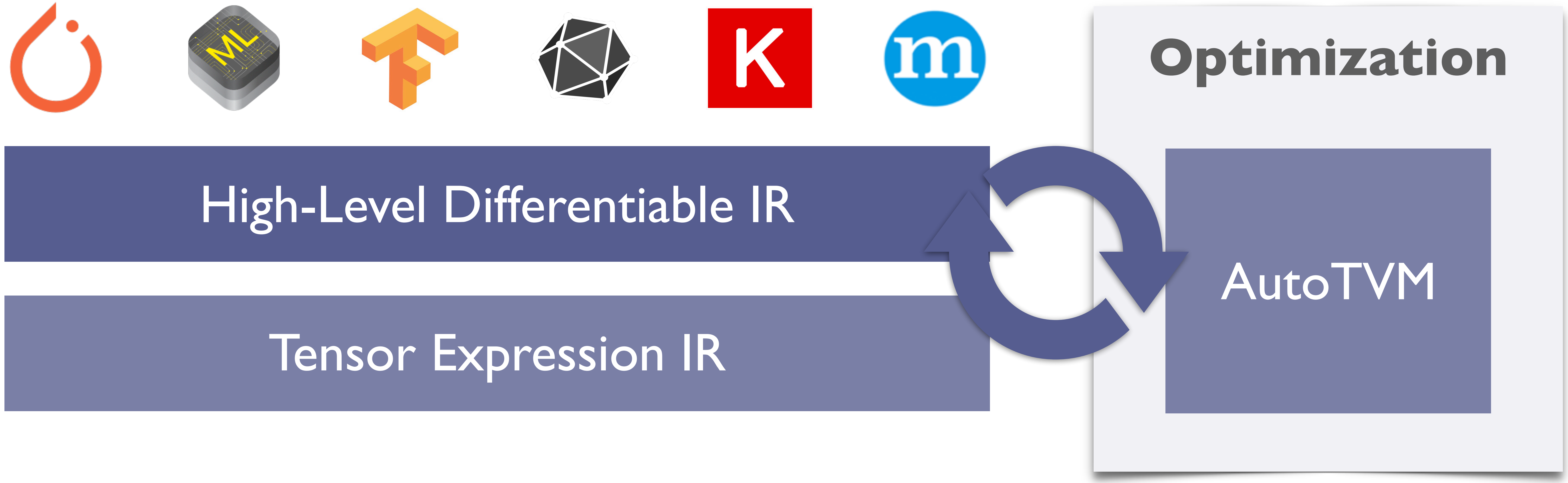
tvm Automated by Machine Learning



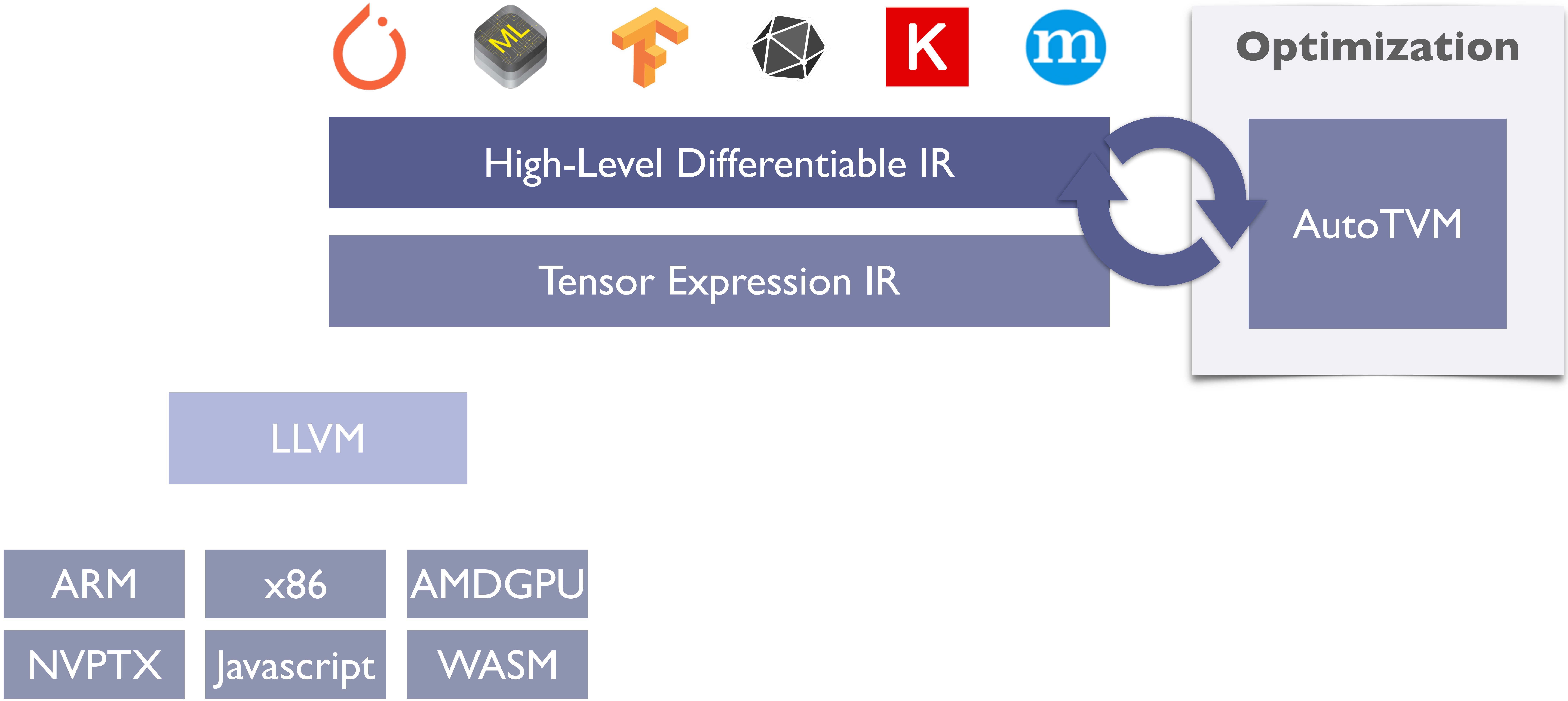
Automated by Machine Learning



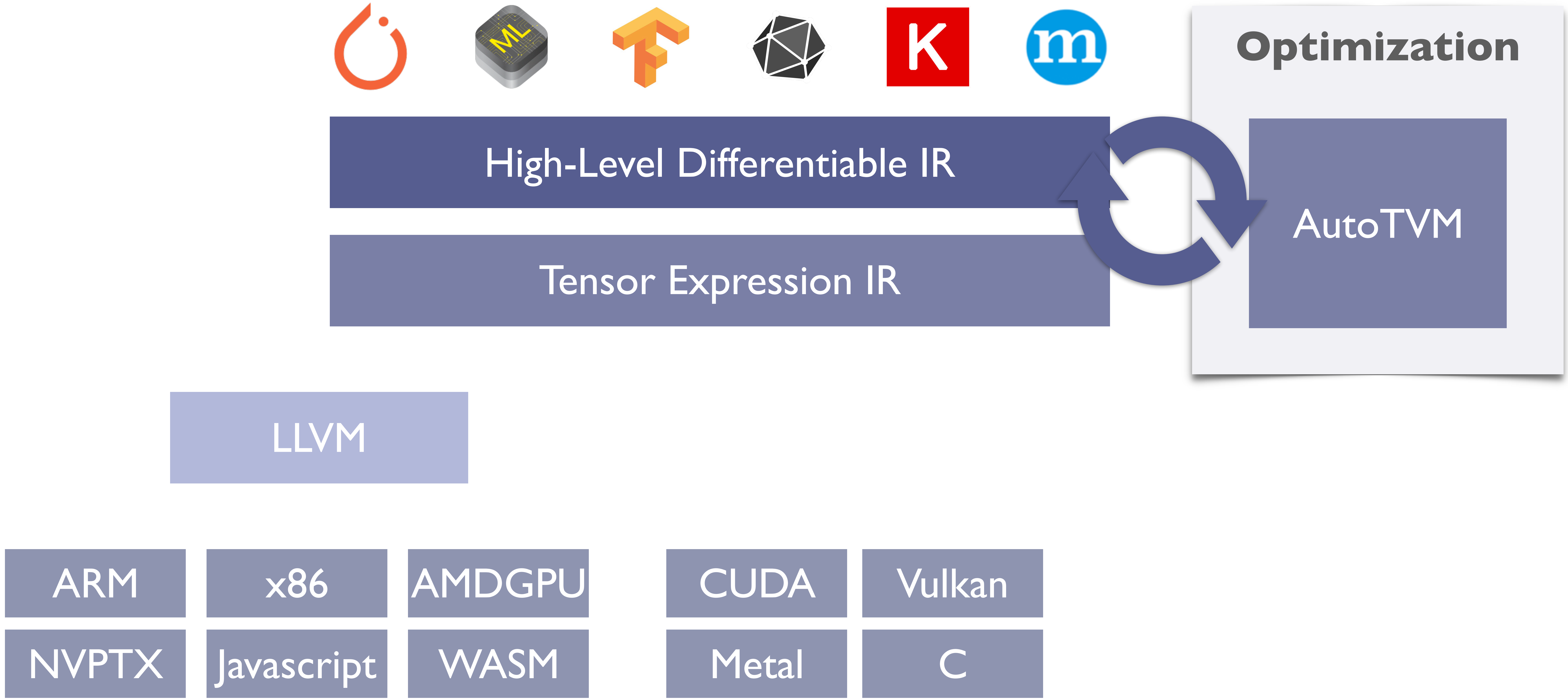
Diverse Hardware backends



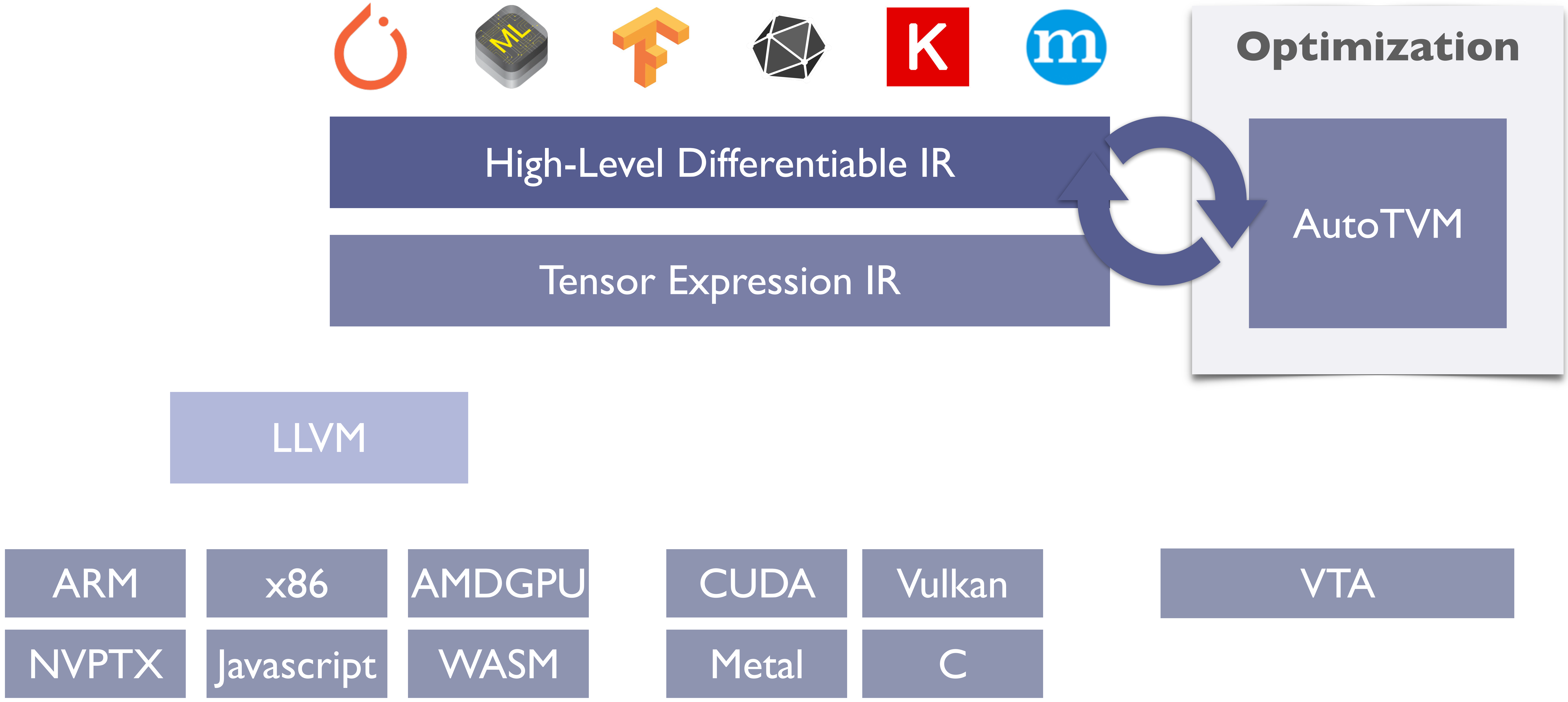
Diverse Hardware backends



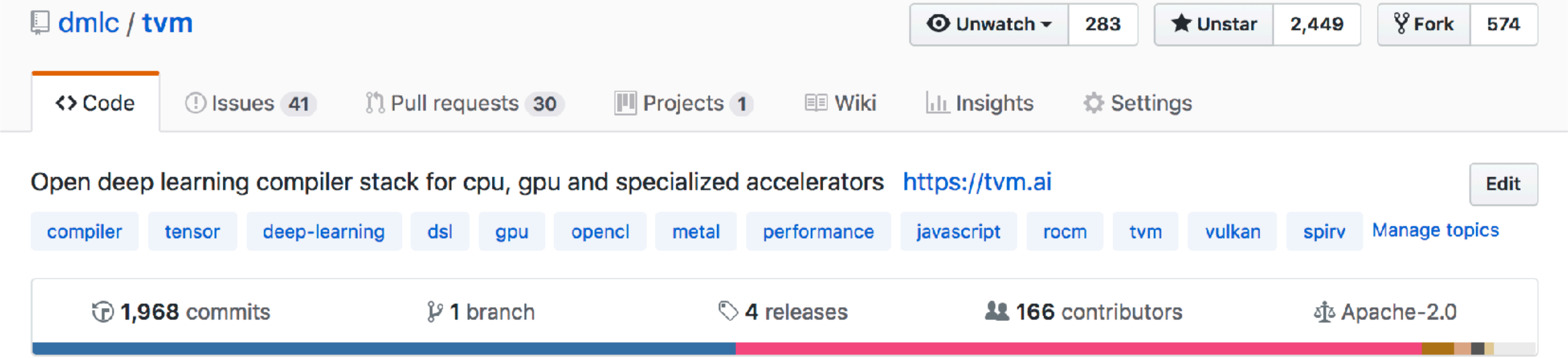
Diverse Hardware backends



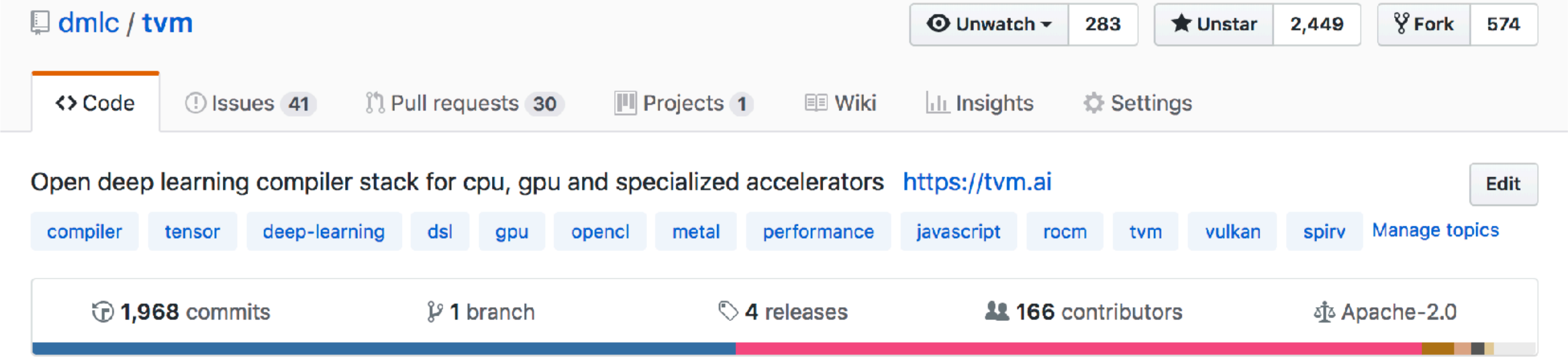
Diverse Hardware backends



TVM Open Source Community



TVM Open Source Community



Apache governance model: grant project ownership by merit.

11 committers, 29 reviewers, 166 contributors.

Contributed by the community, for the community.

Industrial Impact

TVM + AWS

Vin Sharma, Amazon SageMaker Neo

Amazon: vinarm@ | Twitter: ciphr@

How is AWS using TVM?

How is AWS using TVM?

- As a back-end for Apache MXNet
  - To deploy easily onto edge devices
  - To improve performance on target hardware

How is AWS using TVM?

- As a back-end for Apache MXNet
  - To deploy easily onto edge devices
  - To improve performance on target hardware
- As an optimizer for Amazon AI services
  - Amazon Rekognition: To improve end-to-end latency
  - Amazon Alexa: To increase resource efficiency on Echo/Dot

How is AWS using TVM?

- As a back-end for Apache MXNet
  - To deploy easily onto edge devices
  - To improve performance on target hardware
- As an optimizer for Amazon AI services
  - Amazon Rekognition: To improve end-to-end latency
  - Amazon Alexa: To increase resource efficiency on Echo/Dot
- In a tool chain for Amazon Inferentia

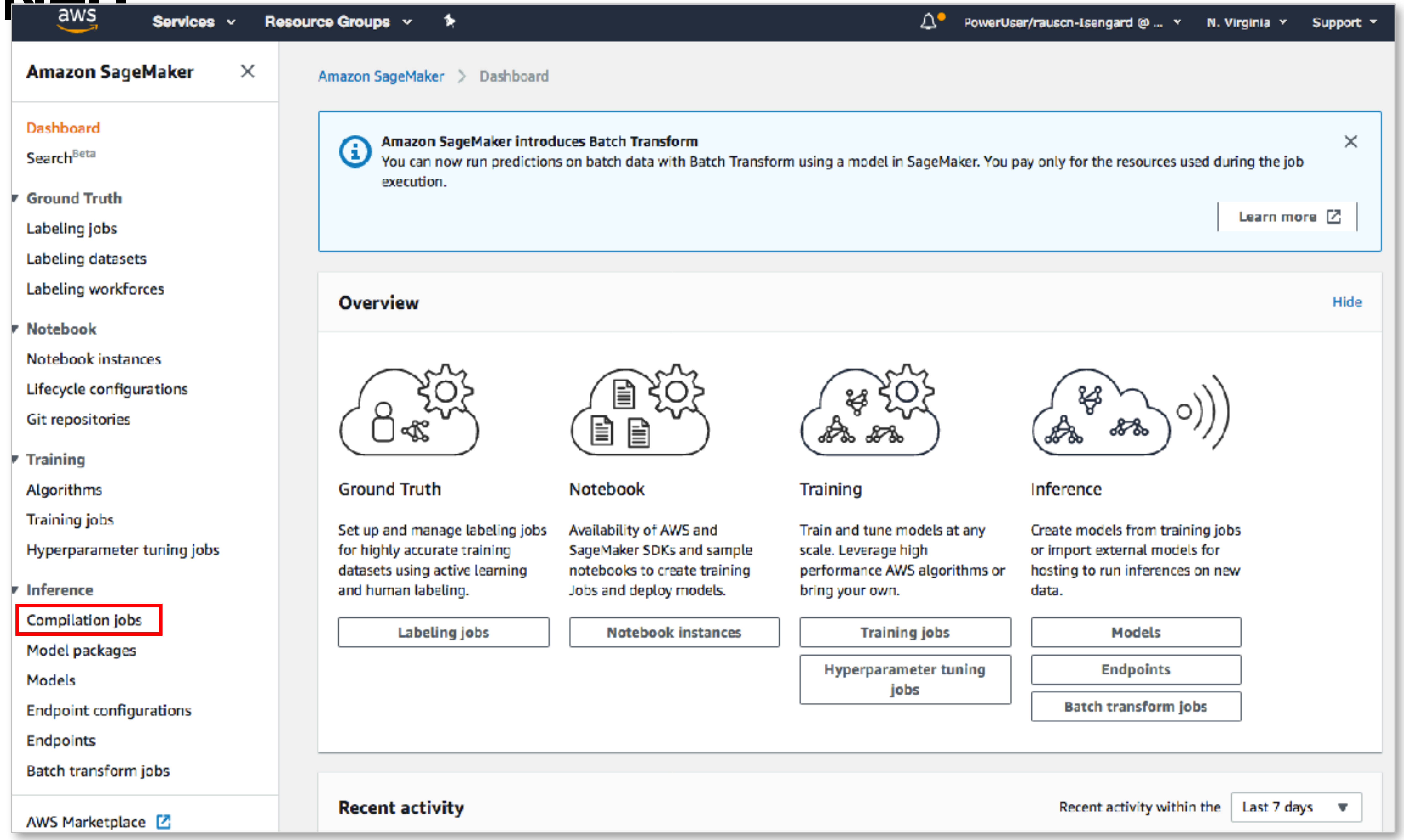
How is AWS using TVM?

We're Hiring!

- As a back-end for Apache MXNet
  - To deploy easily onto edge devices
  - To improve performance on target hardware
- As an optimizer for Amazon AI services
  - Amazon Rekognition: To improve end-to-end latency
  - Amazon Alexa: To increase resource efficiency on Echo/Dot
- In a tool chain for Amazon Inferentia

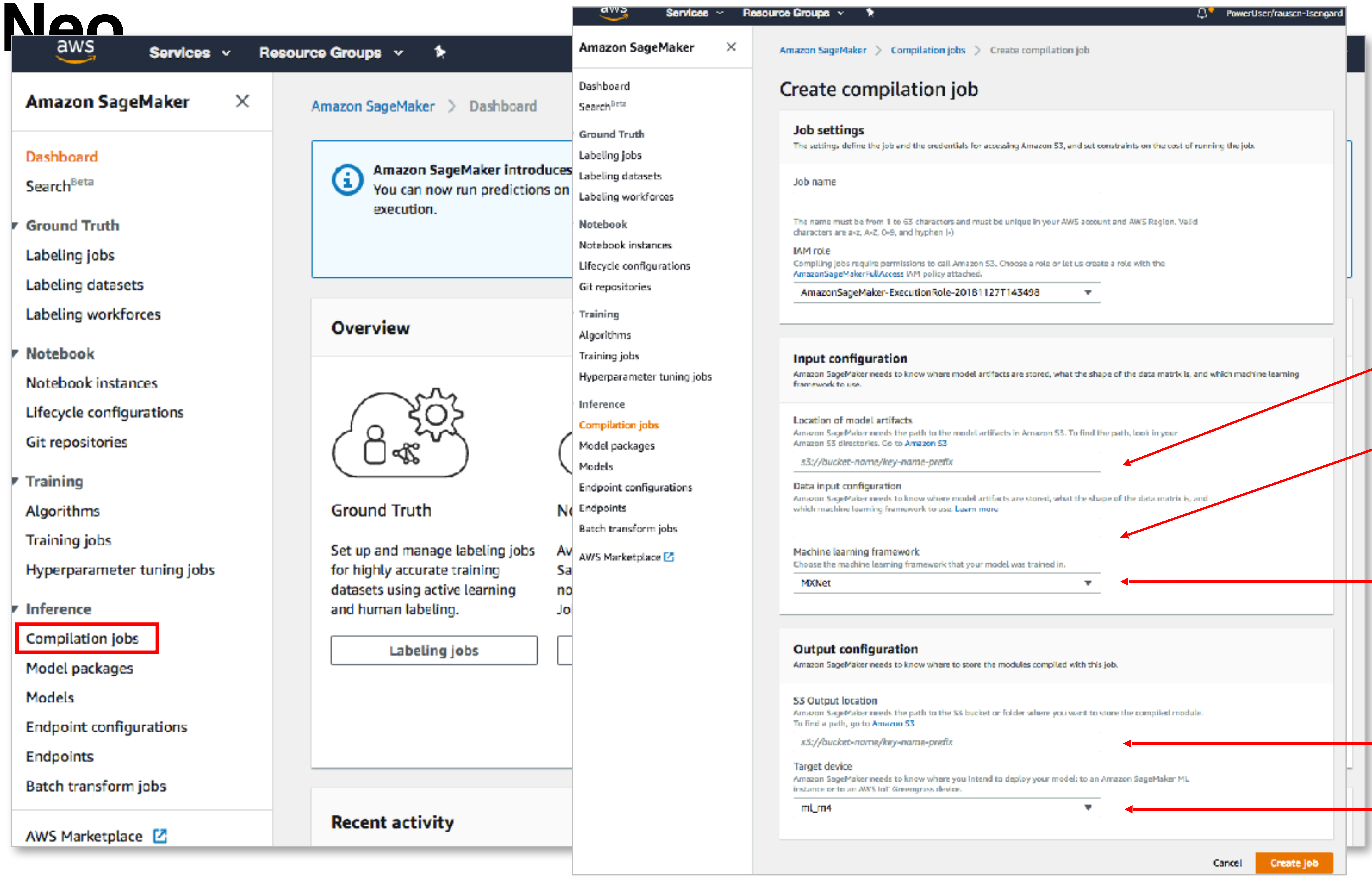
How is AWS enabling adoption of TVM?

In a new service called **Amazon SageMaker Neo**



How is AWS enabling adoption of TVM?

In a new service called **Amazon SageMaker Neo**



Model input files:
MXNet: .json & .params

Name and shape of input node:
{“data”:[1,3,227,277]}

Framework

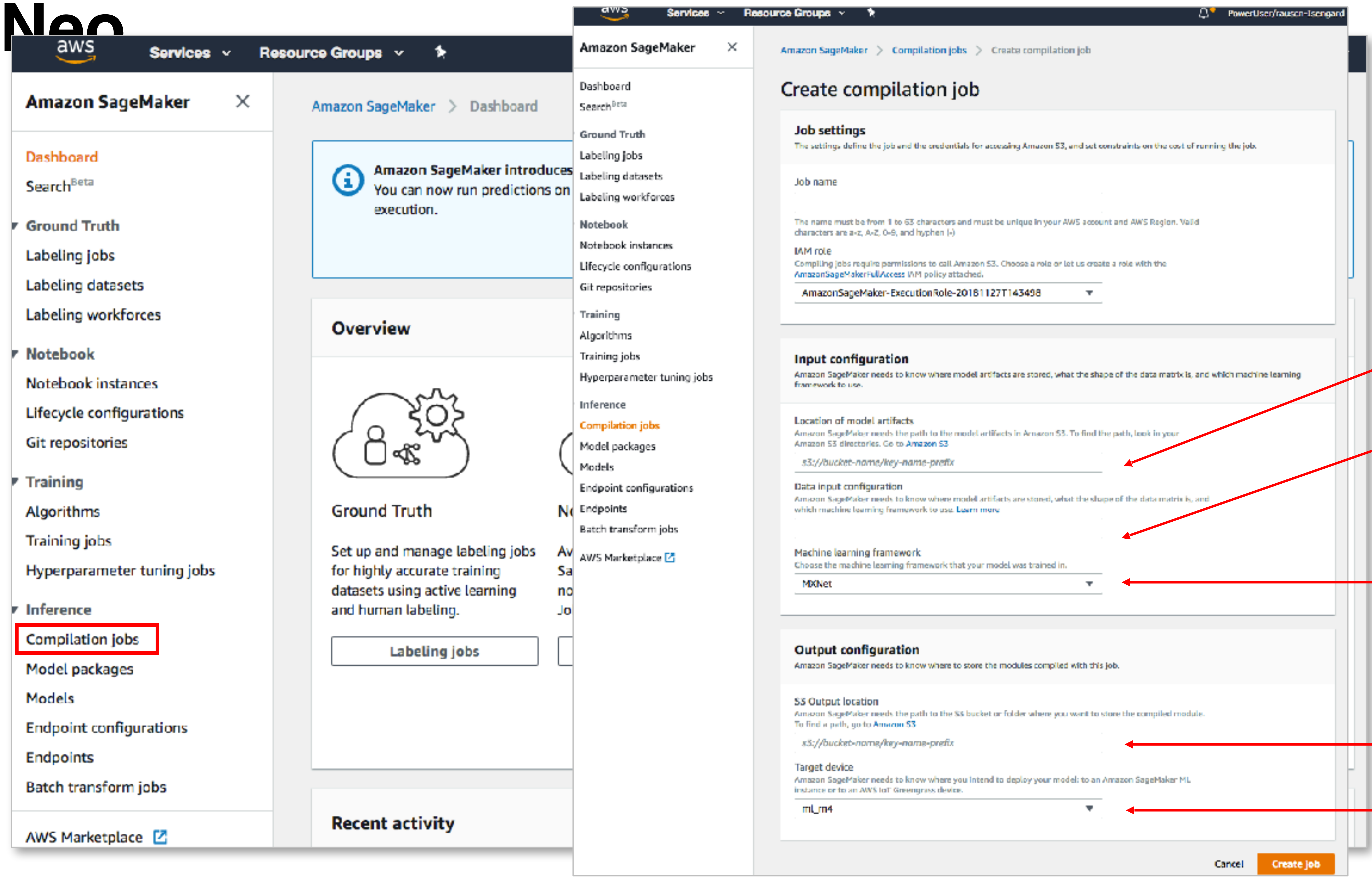
Output Location

Target Platform:
Cloud Instance Type | Edge Device

We're Hiring!

How is AWS enabling adoption of TVM?

In a new service called **Amazon SageMaker Neo**



Model input files:
MXNet: .json & .params

Name and shape of input node:
{"data":[1,3,227,277]}

Framework

Output Location

Target Platform:
Cloud Instance Type | Edge Device

How is AWS contributing to TVM?

Releasing all TVM modifications and enhancements in Neo to open source

- Frameworks: TensorFlow, MXNet, PyTorch, ONNX
- Models: ResNet, VGG, Inception, MobileNet, DenseNet, SqueezeNet
- Operators: Several new ops in NNVM/TVM
- Optimizations: Node Annotation, Graph Partitioning, Ring Buffer, NHWC, Graph Tuning
- Acceleration Library: Nvidia TensorRT
- Hardware: Cross-Compilation to ARM, Intel, Nvidia; More Coming Soon

How is AWS contributing to TVM?

We're
Hiring!

Releasing all TVM modifications and enhancements in Neo to open source

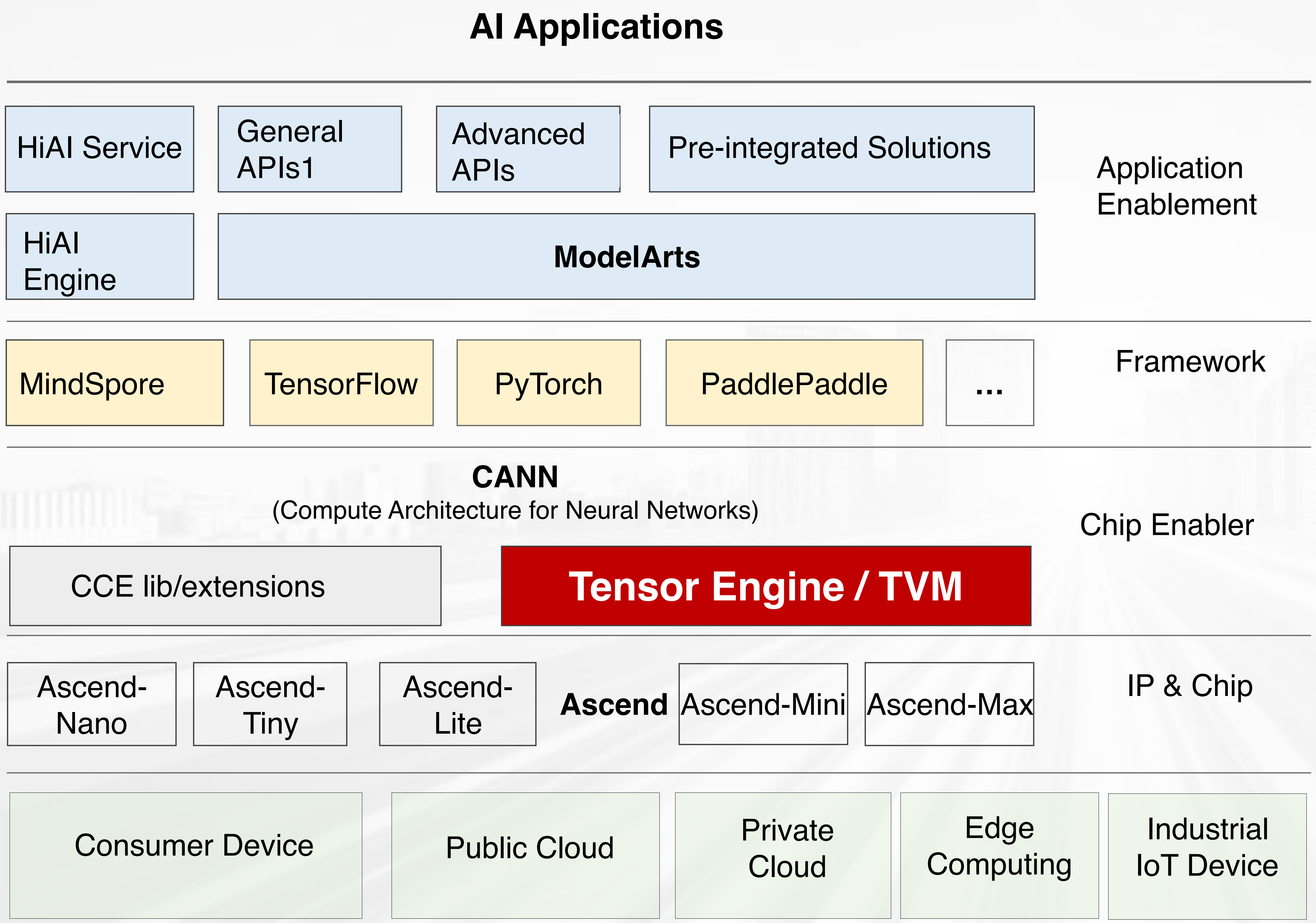
- Frameworks: TensorFlow, MXNet, PyTorch, ONNX
- Models: ResNet, VGG, Inception, MobileNet, DenseNet, SqueezeNet
- Operators: Several new ops in NNVM/TVM
- Optimizations: Node Annotation, Graph Partitioning, Ring Buffer, NHWC, Graph Tuning
- Acceleration Library: Nvidia TensorRT
- Hardware: Cross-Compilation to ARM, Intel, Nvidia; More Coming Soon

sampl

PAUL G. ALLEN SCHOOL
OF COMPUTER SCIENCE & ENGINEERING

Chen Tian, Technical VP

TVM on Huawei’s AI portfolio



Application enabling:
Full-pipeline services(ModelArts), hierarchical APIs, and pre-integrated solutions

MindSpore：
Unified training and inference framework for device, edge, and cloud (both standalone and cooperative)

CANN：
Chip operators library and highly automated operators development toolkit

Ascend：
AI chip series based on unified scalable architecture

How do we use TVM

Frameworks

Caffe

Caffe2

Model Conversion

model execution

TE/TVM

Third-Party Operators

During model conversion we use TE/TVM to customize operators for completeness and performance.

70+ operators are written by TVM， bring us ~3x development efficiency improvement

HUAWEI

Successful Practice with Audi in Level 4 Autonomous Driving

~ A Complete City Commute Record ~

Driving in the evening

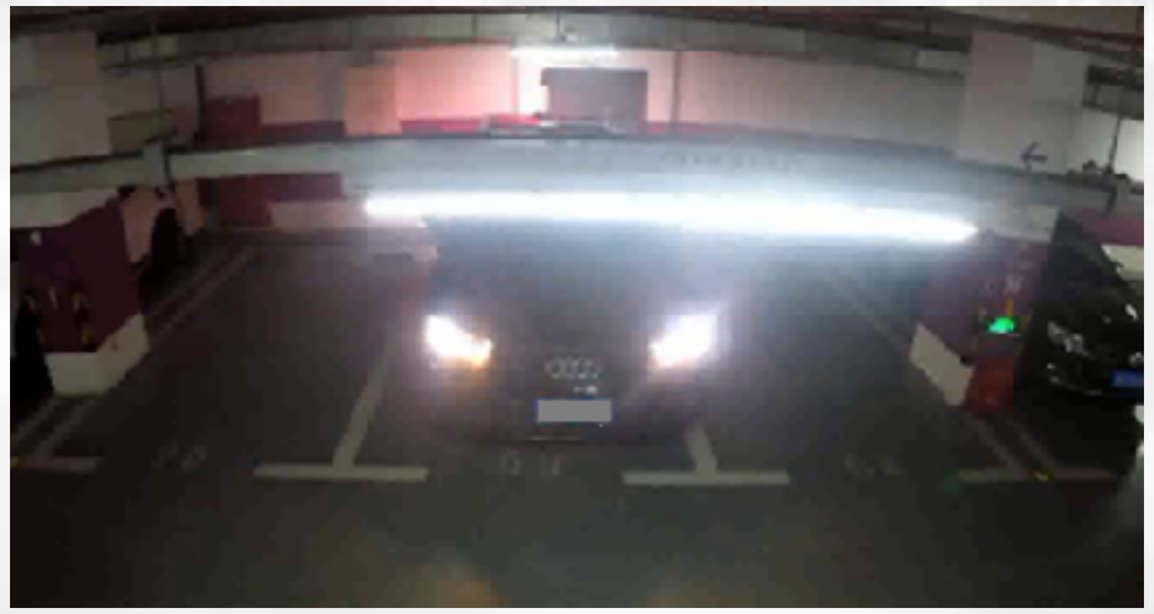
High-speed cruise

Traffic Jam Pilot (TJP)

Traffic light identification

Pedestrian identification

Automatic parking



Joint developed autonomous driving algorithm gains leading scores in industry authoritative KITTI 2D/3D/BEV tests!

HUAWEI

TVM is working on Atlas series product

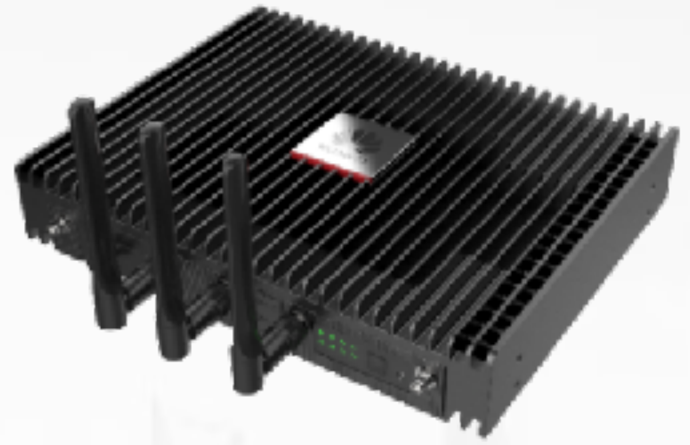
Atlas 200 Developer Kit

- 16 TOPS INT8@24 W
- 1 USB type-C, 2 CCM interfaces, 1 GE network port, 1 SD card slot
- 8 GB memory

Atlas 300 AI Accelerator Card

- 64 TOPS INT8@75 W
- 64-channel HD video real-time analysis and JPEG decoding
- 32 GB memory, 204.8 GB/s memory bandwidth
- PCIe 3.0 x16, half-height half-length card

Atlas 500 AI Edge Station



- Capable of processing 16-channel HD videos in the size of a set-top-box (STB)
- Delivers 4x higher performance over counterparts

Atlas 800 AI Appliance

- Provides optimized AI environment based on the standard framework and programming environment
- Leverages high-performance GPU scheduling algorithms, improving resource utilization by over 15%

Smart Manufacturing

(intelligent quality inspection and flexible manufacturing)

Intelligent Care

(kindergarten and elderly care)

Smart Transportation

(traffic light tuning, intelligent traffic guiding)

Huawei’s Contributions on TVM

8 Contributors：
kun-zh, sgrechanik-h, libing4752, derisavi-huawei, solin319, ehsanmok, gaoxiong-1, jiacunjiang1215

4 Reviewers：
Srkreddy1238 , ParikshеetPinjari909 , siju-Samuel , Xqdan

We are working on：
1.Huawei Ascend ASIC support.
2.Front end to support Darknet, ONNX.
3.Optimization on Auto-TVM, IR extensions.
4.Tensorize, cache read/write, access_ptr API.

In the future we will try to：
1.Codegen for fused operators.
2.NLP support.
3.More optimization.
4.Training Operators.

Meghan Cowan

PAUL G. ALLEN SCHOOL
OF COMPUTER SCIENCE & ENGINEERING

sampl

VGG11 on Raspberry Pi 3B

TensorflowLite
32bit fp
66% top-1ImageNet accuracy
1.42 fps

VGG11 on Raspberry Pi 3B

Trained
binarized model

Operators
implemented with TVM

TensorflowLite
32bit fp
66% top-1ImageNet accuracy
1.42 fps

VGG11 on Raspberry Pi 3B

Trained
binarized model

Operators
implemented with TVM

TensorflowLite
32bit fp
66% top-1ImageNet accuracy
1.42 fps

TVM
2-bit activation 1-bit weight
62% top-1 ImageNet accuracy
4.67 fps

Further down the stack…

Thierry Moreau

PAUL G. ALLEN SCHOOL
OF COMPUTER SCIENCE & ENGINEERING

sampl

Open Source Stack Overview

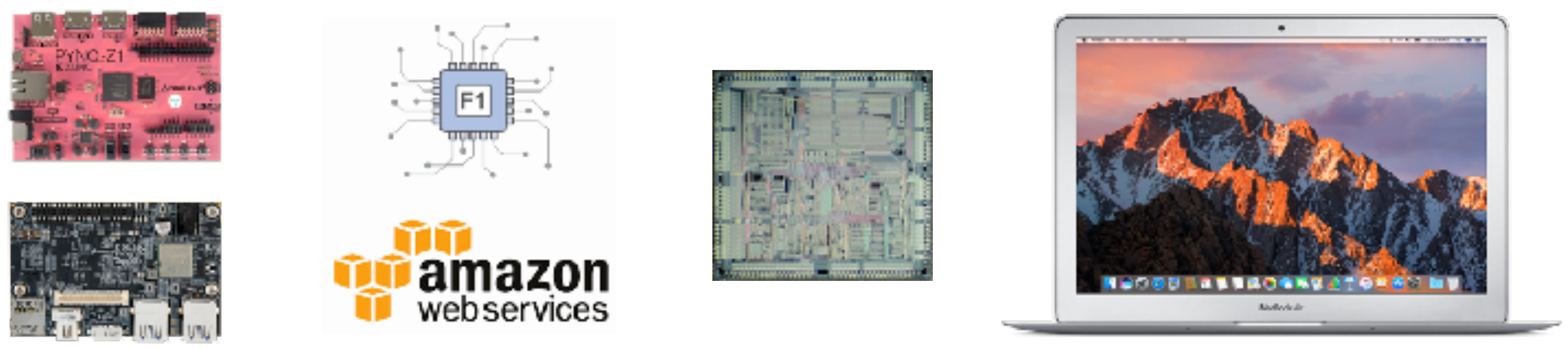
High-Level Differentiable IR

Tensor Expression IR

VTA Runtime & JIT Compiler

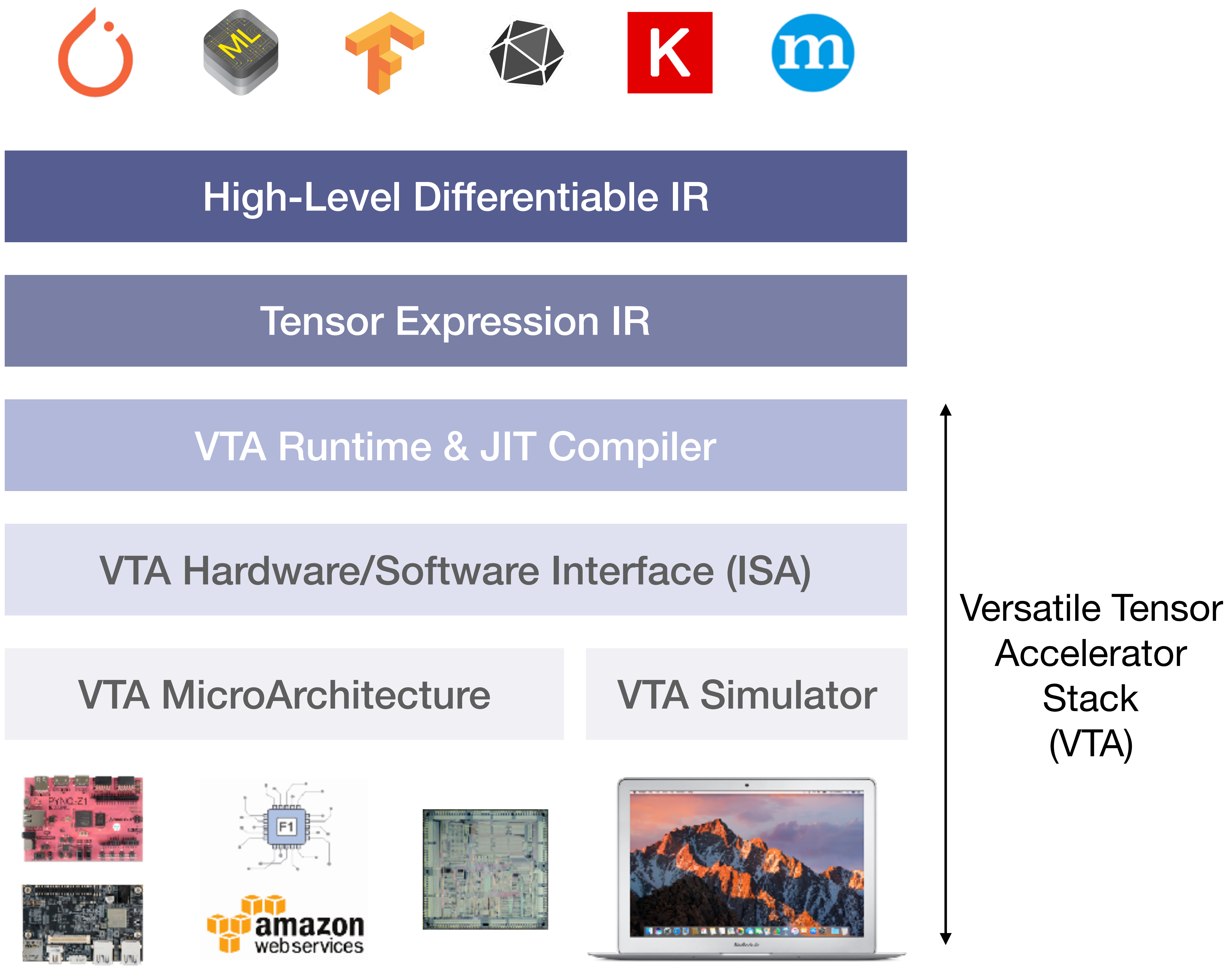
VTA Hardware/Software Interface (ISA)

VTA MicroArchitecture | VTA Simulator



Versatile Tensor Accelerator Stack (VTA)

Open Source Stack Overview



VTA Backends

- **Simulator**: out-of-the-box testing to write compiler passes

Open Source Stack Overview

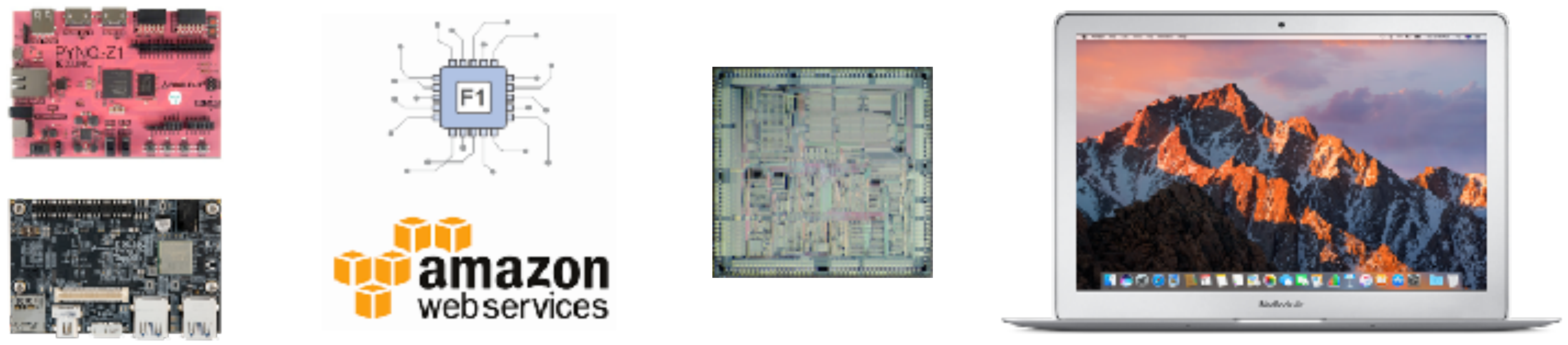
High-Level Differentiable IR

Tensor Expression IR

VTA Runtime & JIT Compiler

VTA Hardware/Software Interface (ISA)

VTA MicroArchitecture | VTA Simulator

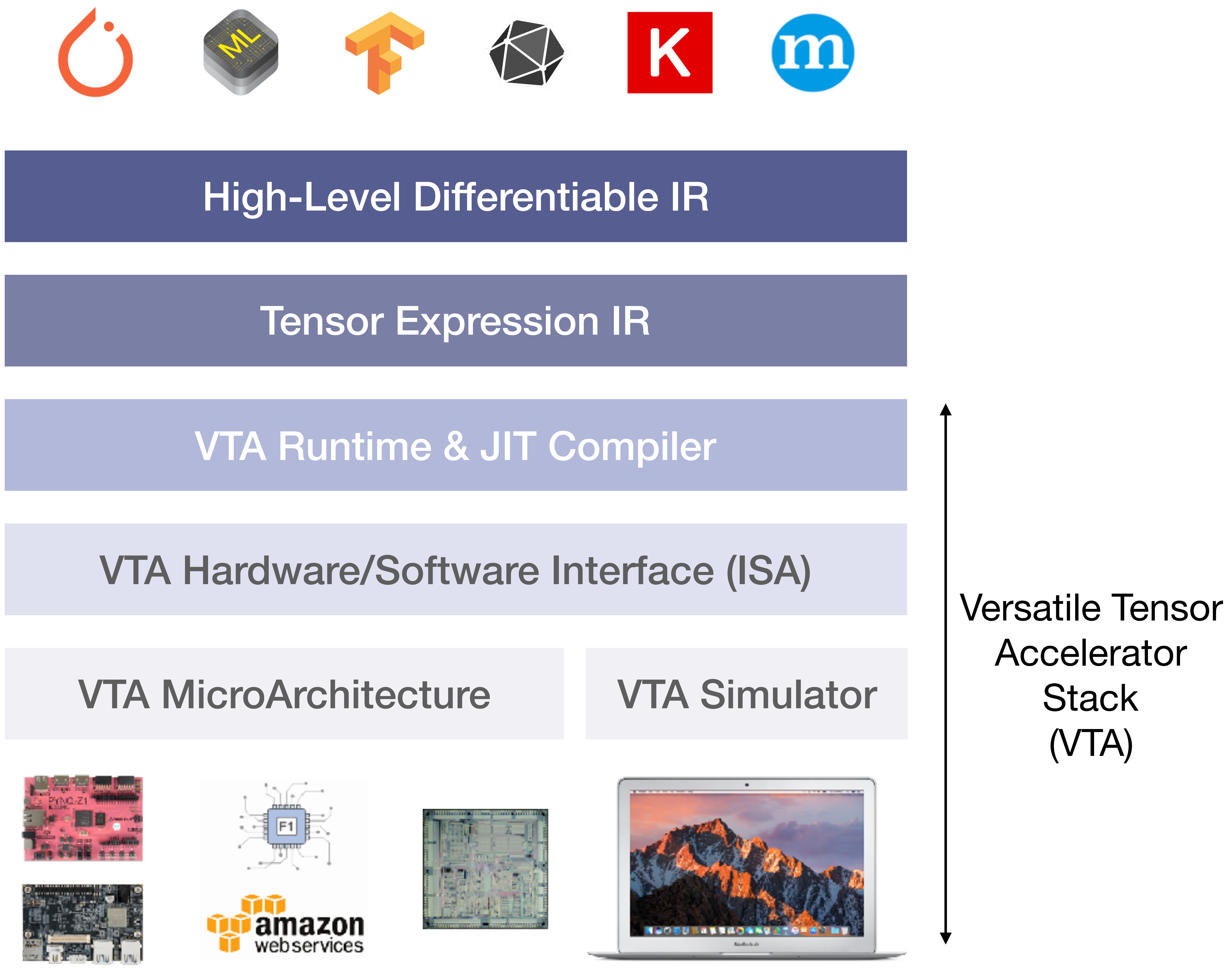


Versatile Tensor Accelerator Stack (VTA)

VTA Backends

- **Simulator**: out-of-the-box testing to write compiler passes
- **FPGA**: fast design iteration, quick deployment, flexibility

Open Source Stack Overview

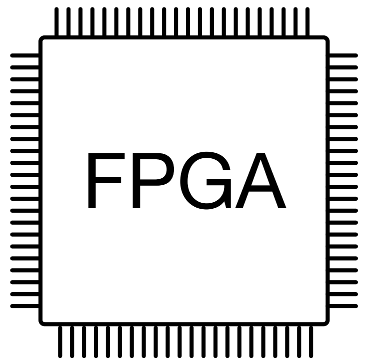


VTA Backends

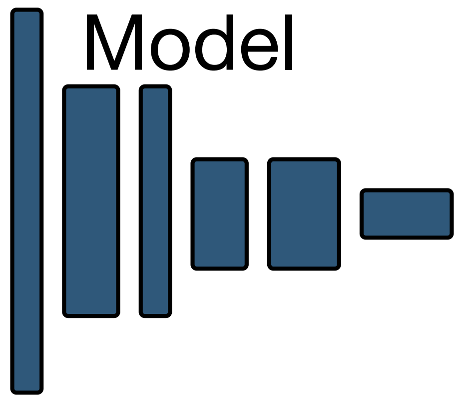
- **Simulator**: out-of-the-box testing to write compiler passes
- **FPGA**: fast design iteration, quick deployment, flexibility
- **ASIC**: industrial-strength efficiency

Hardware Exploration with VTA

HW / SW Constraints



BRAMs
DRAM channels
logic resources



batch size
data types
channel width

Hardware Exploration with VTA

HW / SW Constraints

FPGA

- # BRAMs
- DRAM channels
- logic resources

Model

- batch size
- data types
- channel width

VTA Design Space

Architecture Knobs

- GEMM Intrinsic: e.g. (1,32) x (32,32) vs. (4,16) x (16,16)
- # of units in tensor ALU : e.g. 32 vs. 16
- BRAM allocation between buffers, register file, micro-op cache

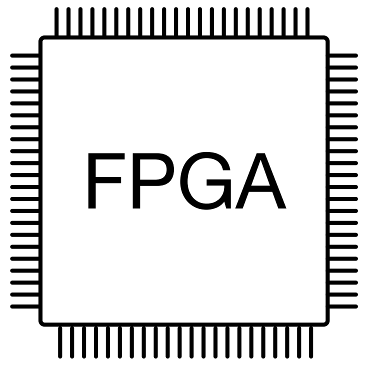
Circuit Knobs

- Circuit Pipelining: e.g. for GEMM core between [11, 20] stages
- PLL Frequency Sweeps: e.g. 250 vs. 300 vs. 333MHz

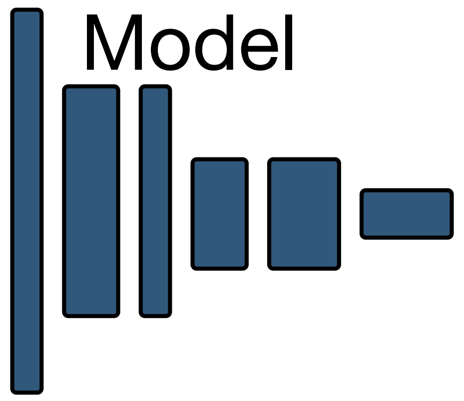
1000s

Hardware Exploration with VTA

HW / SW Constraints



BRAMs
DRAM channels
logic resources



batch size
data types
channel width

VTA Design Space

Architecture Knobs

- GEMM Intrinsic: e.g. (1,32) x (32,32) vs. (4,16) x (16,16)
- # of units in tensor ALU : e.g. 32 vs. 16
- BRAM allocation between buffers, register file, micro-op cache

Circuit Knobs

- Circuit Pipelining: e.g. for GEMM core between [11, 20] stages
- PLL Frequency Sweeps: e.g. 250 vs. 300 vs. 333MHz

1000s

VTA Candidate Designs

- #1 Design AAA @ 307GOPs
- #2 Design BBB @ 307GOPs
- #3 Design CCC @ 307GOPs
- #4 Design DDD @ 256GOPs

Needs to pass place & route
and pass timing closure

~10

Schedule Exploration with VTA

VTA Candidate Designs

- #1 Design AAA @ 307GOPs
- #2 Design BBB @ 307GOPs
- #3 Design CCC @ 307GOPs
- #4 Design DDD @ 256GOPs

Needs to pass place & route and pass timing closure

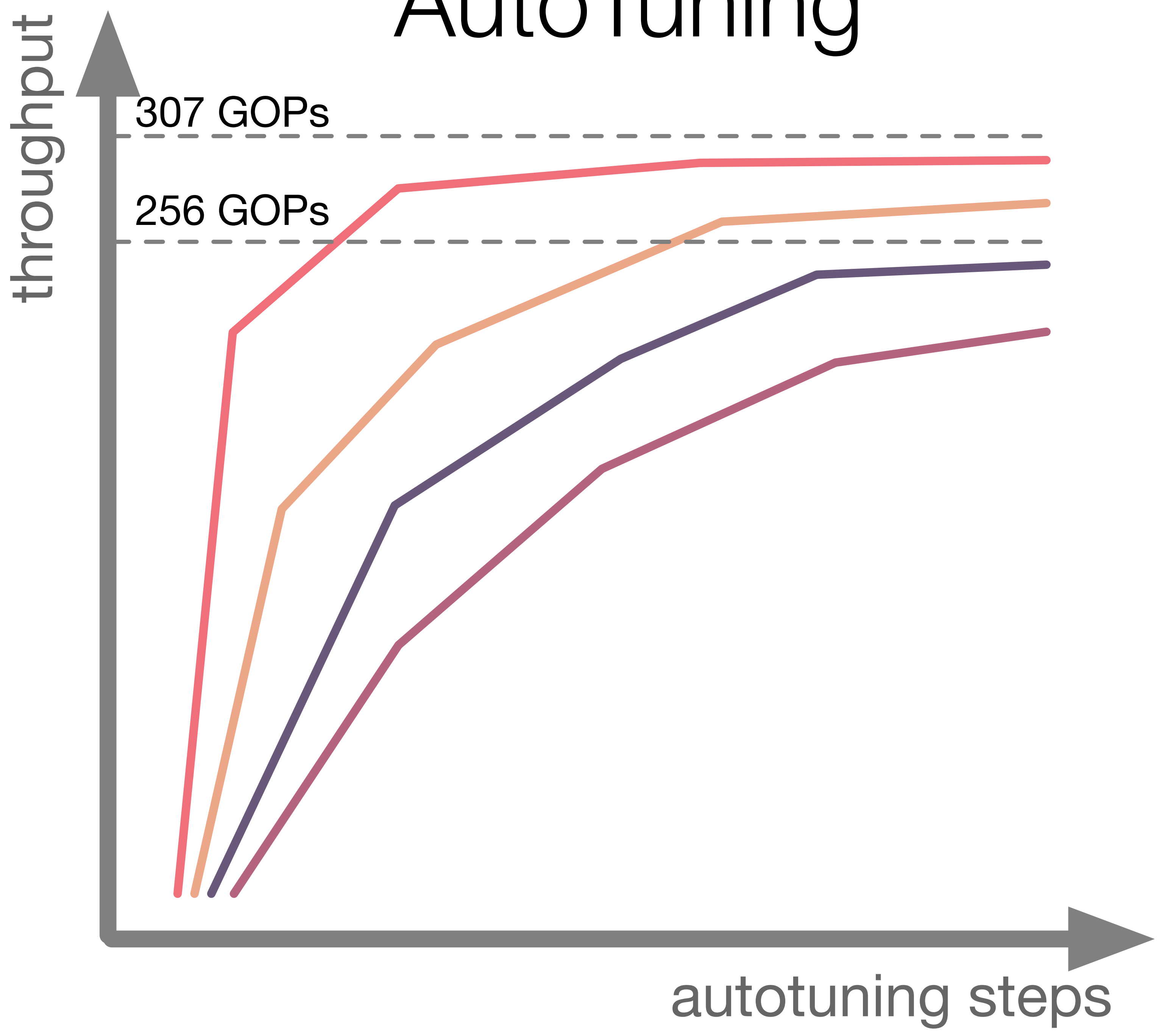
Schedule Exploration with VTA

VTA Candidate Designs

- #1 Design AAA @ 307GOPs
- #2 Design BBB @ 307GOPs
- #3 Design CCC @ 307GOPs
- #4 Design DDD @ 256GOPs

Needs to pass place & route
and pass timing closure

Operator Performance
AutoTuning



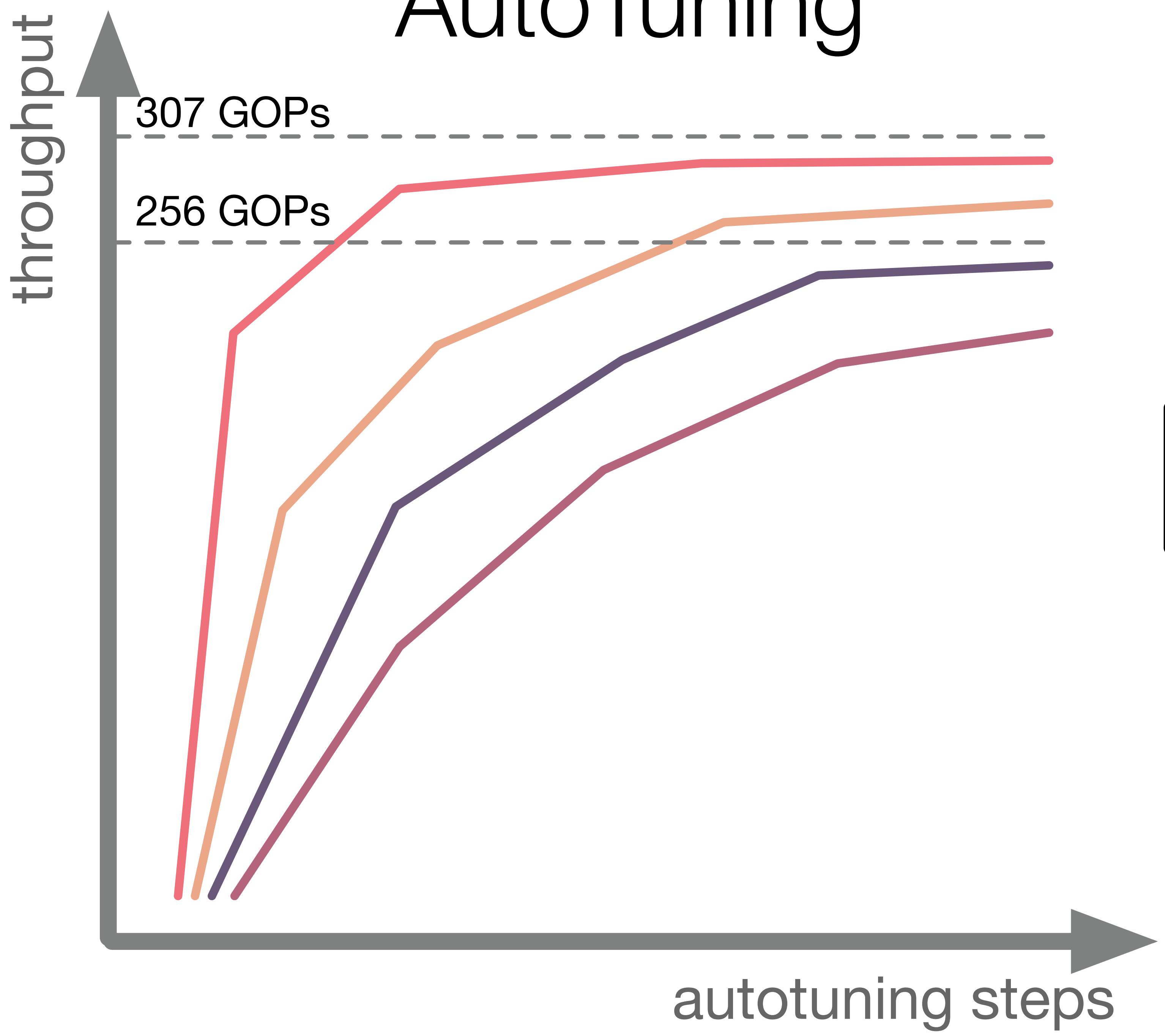
Schedule Exploration with VTA

VTA Candidate Designs

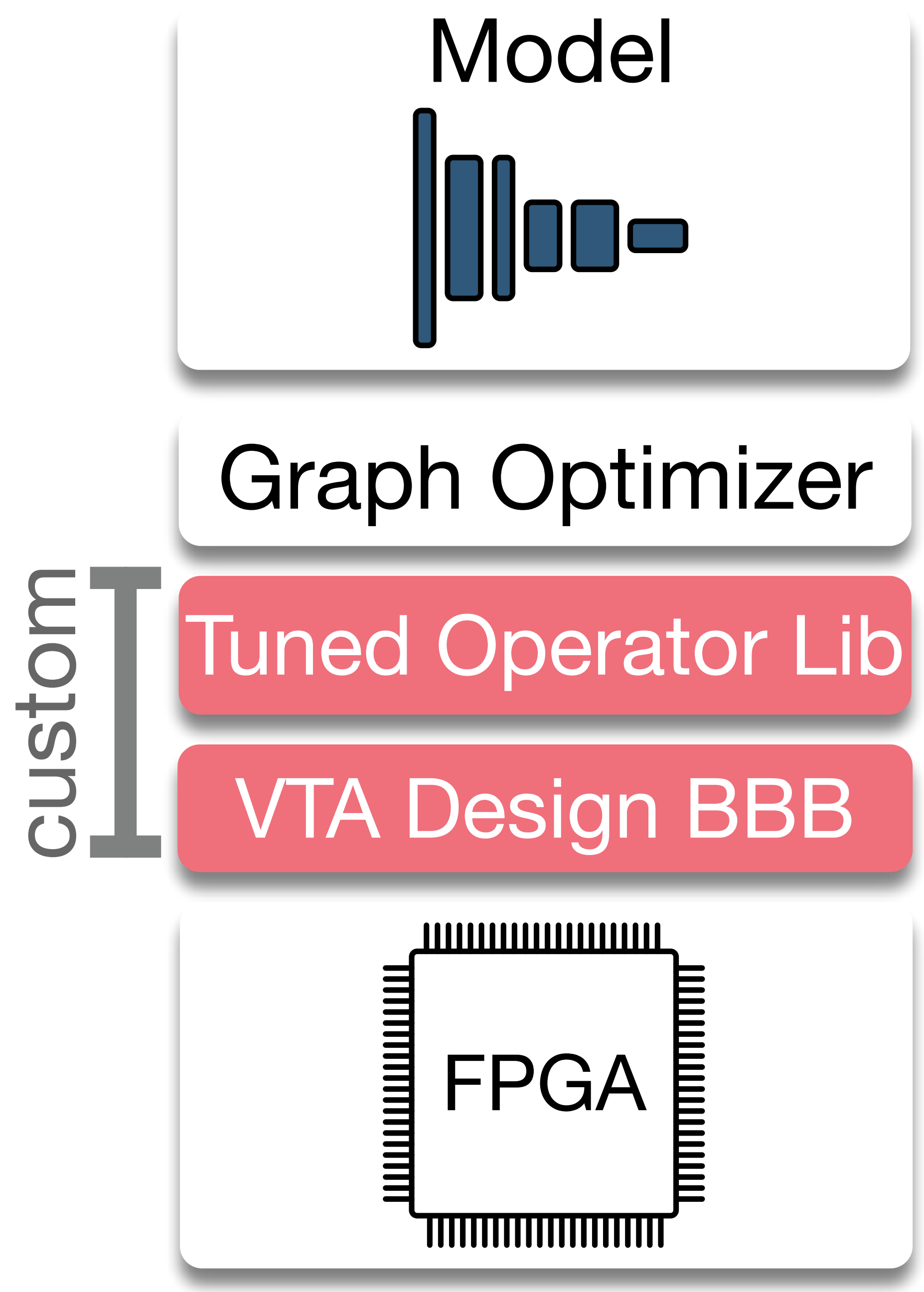
- #1 Design AAA @ 307GOPs
- #2 Design BBB @ 307GOPs
- #3 Design CCC @ 307GOPs
- #4 Design DDD @ 256GOPs

Needs to pass place & route
and pass timing closure

Operator Performance AutoTuning

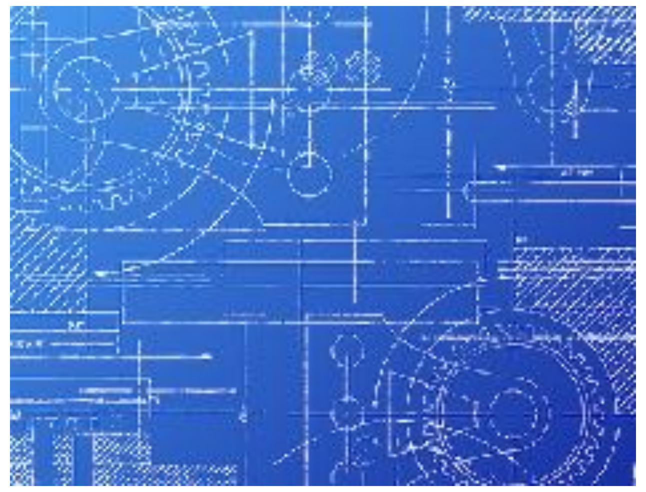


Deliverable



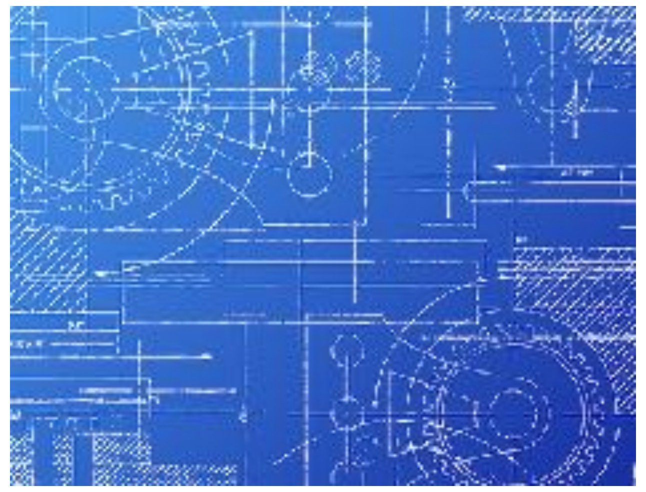
TVM+VTA Stack Goals

TVM+VTA Stack Goals



- Blue-print for a complete deep learning acceleration stack

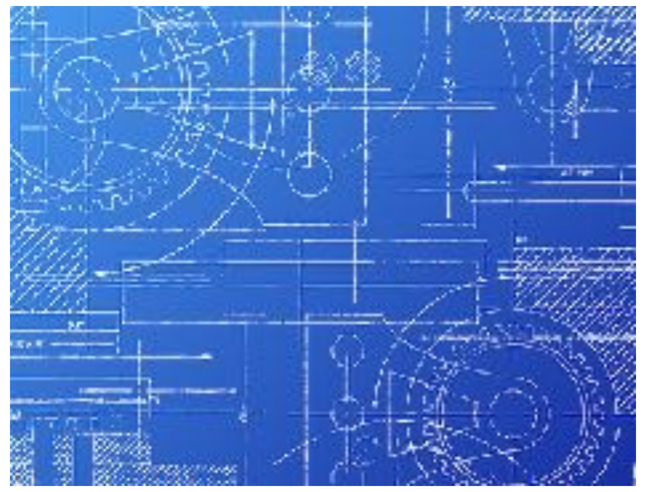
TVM+VTA Stack Goals



- Blue-print for a complete deep learning acceleration stack

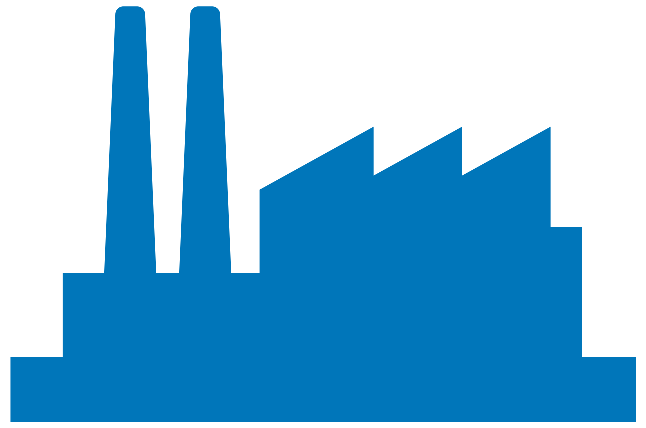
- Experimentation framework for cross-stack deep learning optimizations

TVM+VTA Stack Goals



- Blue-print for a complete deep learning acceleration stack

- Experimentation framework for cross-stack deep learning optimizations



- Open-source community for industrial-strength deep learning acceleration

Carlos Guestrin

PAUL G. ALLEN SCHOOL
OF COMPUTER SCIENCE & ENGINEERING

sampl

Training Deep Learning Models with TVM

Jared Roesch

PAUL G. ALLEN SCHOOL
OF COMPUTER SCIENCE & ENGINEERING

sampl

Model

High-Level Differentiable IR

Tensor Expression IR

LLVM, CUDA, Metal

VTA

Edge FPGA

Cloud FPGA

ASIC

Standalone inference deployment

Model

High-Level Differentiable IR

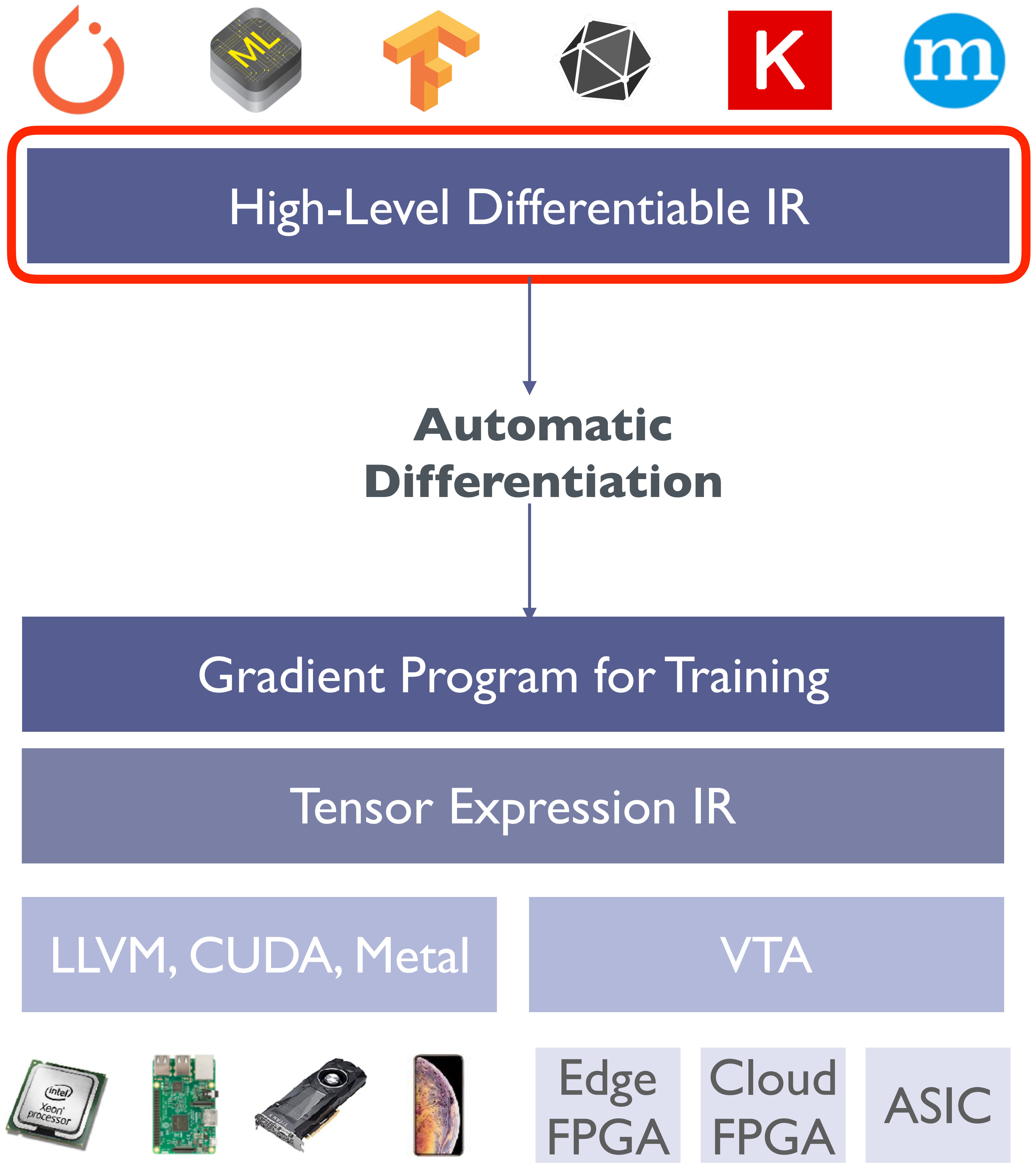
Tensor Expression IR

LLVM, CUDA, Metal | VTA

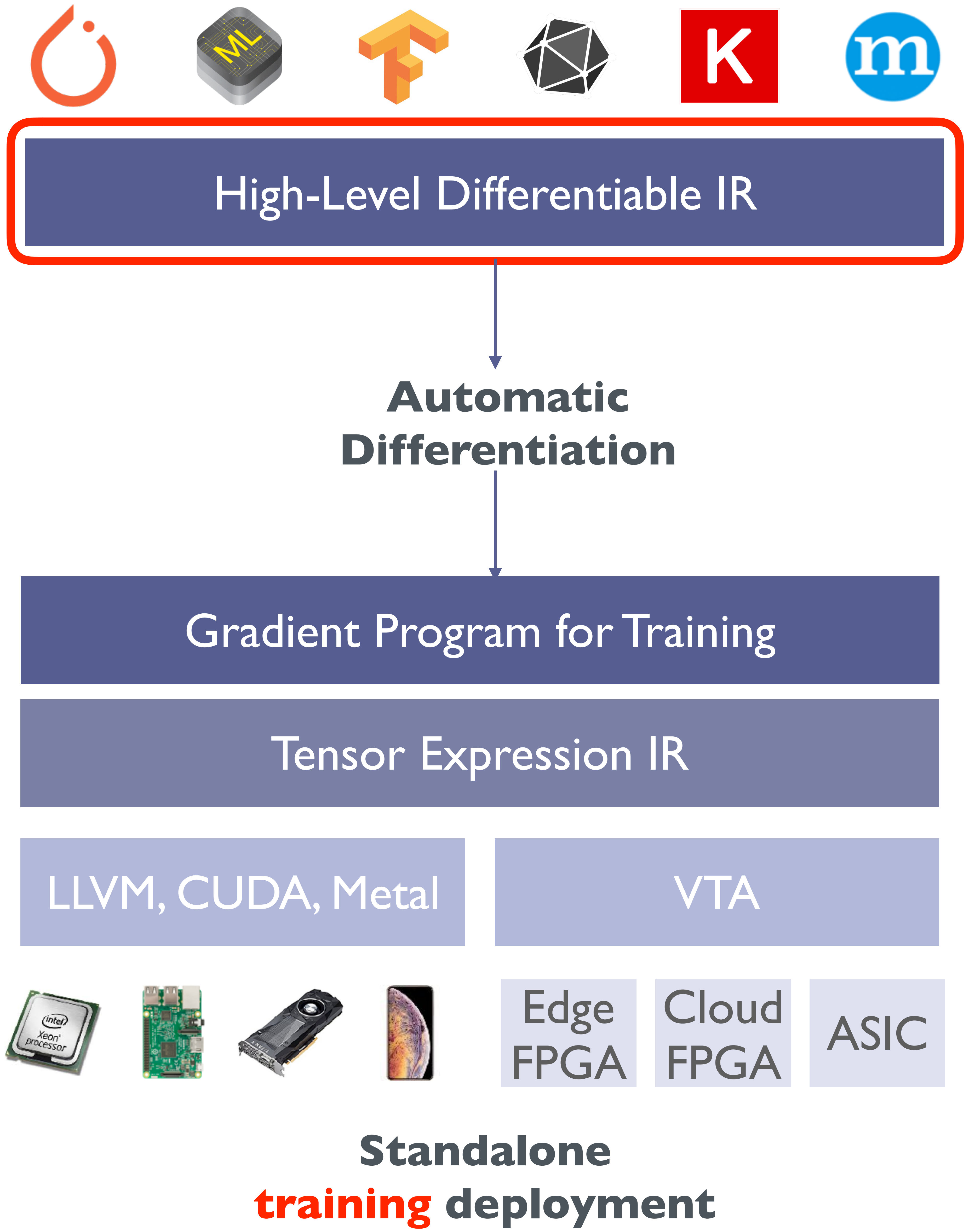
Edge FPGA | Cloud FPGA | ASIC

Standalone inference deployment

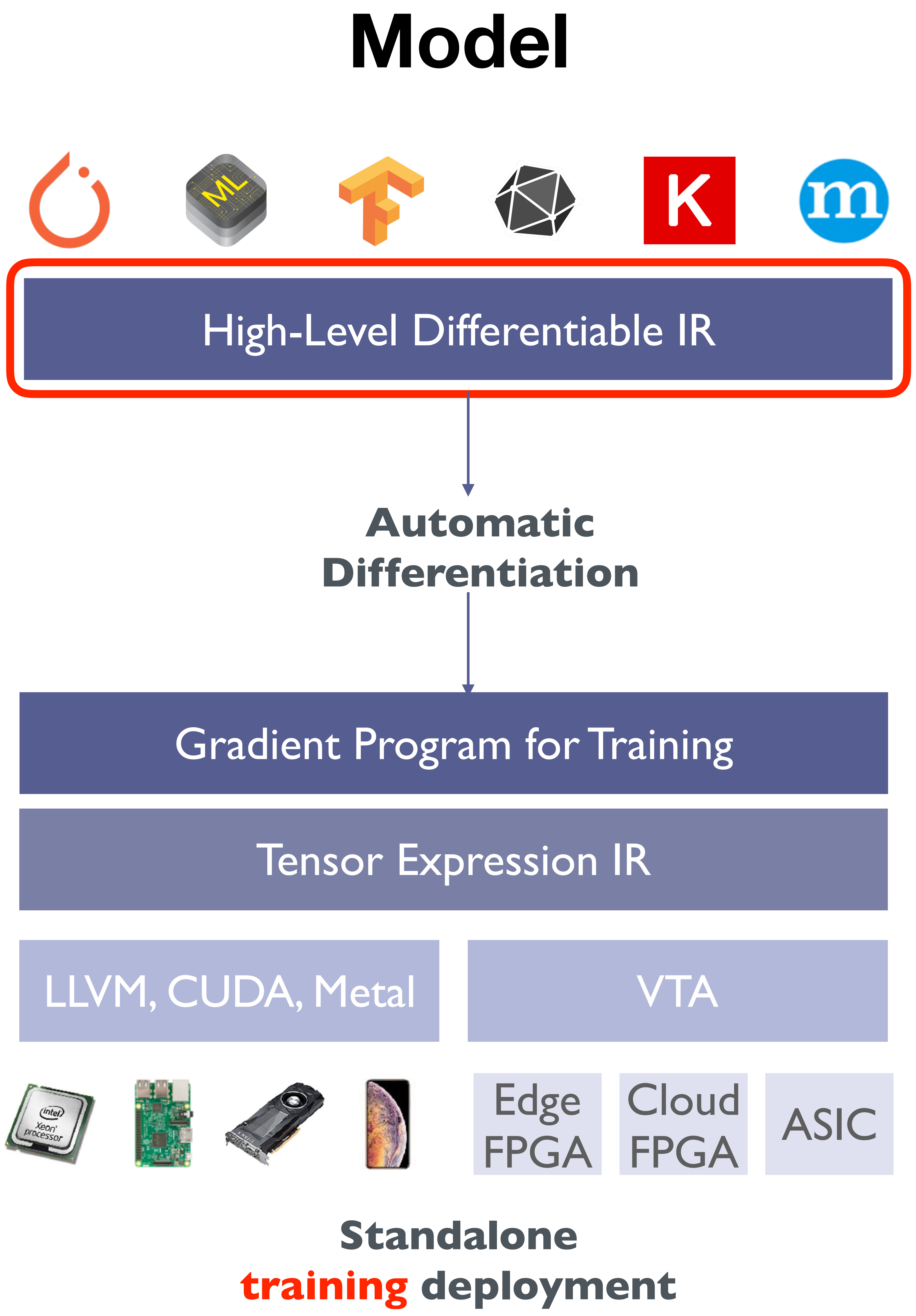
Model


High-Level Differentiable IR
Automatic Differentiation
Gradient Program for Training
Tensor Expression IR
LLVM, CUDA, Metal
VTA
Edge FPGA
Cloud FPGA
ASIC

Model



High-Level Differentiable IR
Automatic Differentiation
Gradient Program for Training
Tensor Expression IR
LLVM, CUDA, Metal
VTA
Edge FPGA
Cloud FPGA
ASIC
Standalone training deployment



- Automatic generation of gradient programs
- Support for customized data types and FPGA training
- Support for distributed execution, and integration with technology such as PHub (see Liang's talk).

More details on the Relay talk later today!

Road ahead…

On the horizon…

Training

Automation

Hardware

On the horizon…

Training

AutoDiff with Relay

Training on-device

Tradeoff accuracy/throughput/Joules

Automation

Hardware

On the horizon…

Training	AutoDiff with Relay	Training on-device	Tradeoff accuracy/ throughput/Joules
Automation	Auto quantization	Full-program optimization	Automated HW design
Hardware			

On the horizon…

Training	AutoDiff with Relay	Training on-device	Tradeoff accuracy/ throughput/Joules
Automation	Auto quantization	Full-program optimization	Automated HW design
Hardware	VTA Chisel design	ASIC flow	Training on VTA

Big THANKS to our sponsors!

Time	Session	Speakers
9:00	Keynote, TVM Overview,TVM @ Amazon	Keynote (SAMPL, Apple, Amazon, Huawei) TVM Overview – Tianqi Chen, UW Deep Learning Compilation at Amazon – Yida Wang, Amazon
11:05	*Break*	
11:25	Automation, HW Specialization, Security	AutoTVM & Device Fleet – Eddie Yan, UW VTA Open Source Deep Learning Accelerator – Thierry Moreau, UW Secure Enclaves for Deep Learning – Nick Hynes, UC Berkeley/Oasis Labs
12:30	*Boxed lunches*	
13:30	Training, Programming Systems, Hardware	Kunle Olukotun/Raghu Prabhakar, Stanford & SambaNova Machine Programming – Justin Gottschlich, Intel PlaidML Stripe: Polyhedral IR + Model-guided Optimization – Brian Retford, Intel The Relay Differentiable IR for TVM – Jared Roesch, UW Scalable Distributed Training with Parameter Hub – Liang Luo, UW The HammerBlade ML Supercomputer – Michael Taylor, UW
15:20	*Break, contributors meetup*	
15:50	Compilers, FPGAs	Andrew Tulloch, Facebook Graham Schelle, Xilinx
16:30	Lightning talks	
17:35	Community formation	Markus Weimer, Microsoft and Apache Software Foundation
18:10	Social (food, drinks)	
20:00	*adjourn*	