2nd TVM and Deep Learning Compilation Conference

tvm

sampl

PAUL G.
ALLEN
SCHOOL
W

December 5, 2019

Luis Ceze

PAUL G. ALLEN SCHOOL
OF COMPUTER SCIENCE & ENGINEERING

sampl

OctoML

Welcome to the ~~1st~~ 2nd TVM and Deep Learning Compilation Conference!

Welcome to the ~~1st~~ 2nd TVM and Deep Learning Compilation Conference!

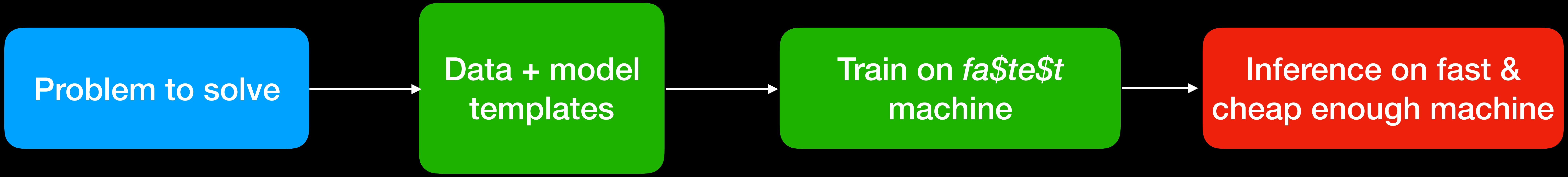
200+ ppl!

Welcome to the ~~1st~~ 2nd TVM and Deep Learning Compilation Conference!

200+ ppl!

2020 →

Machine learning era:



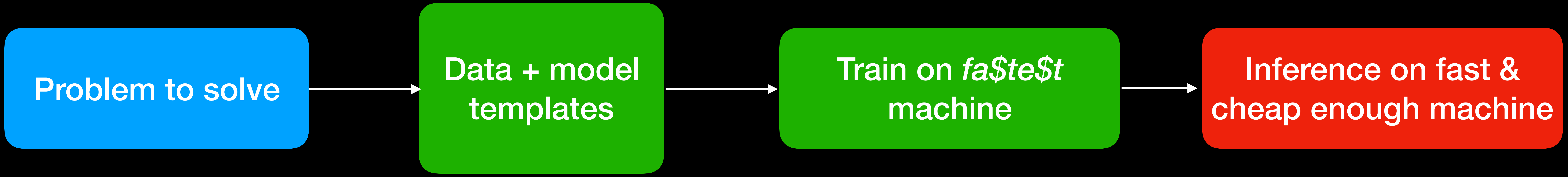
Problem to solve

Data + model templates

Train on *fa$te$t* machine

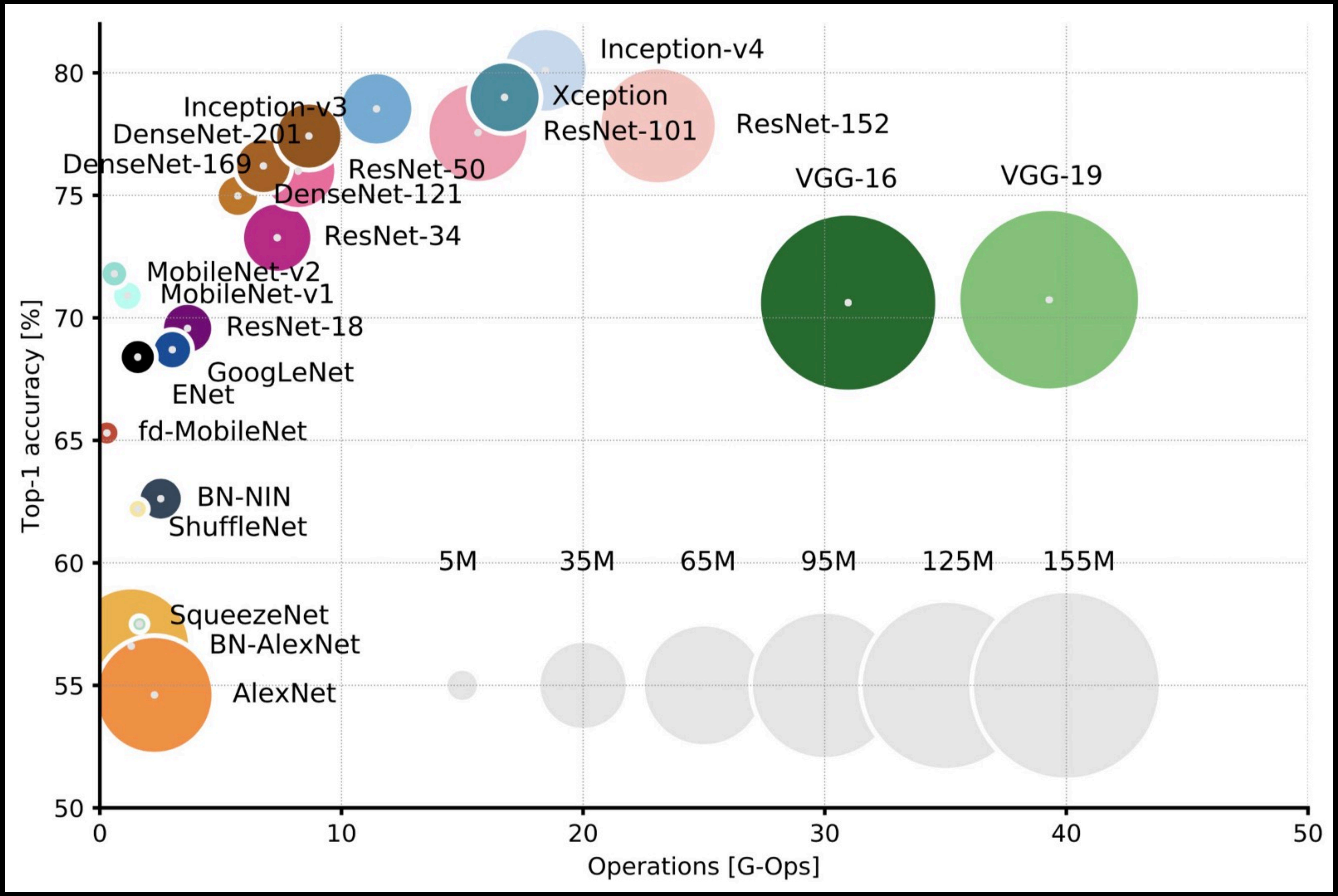
Inference on fast & cheap enough machine

Machine learning era:



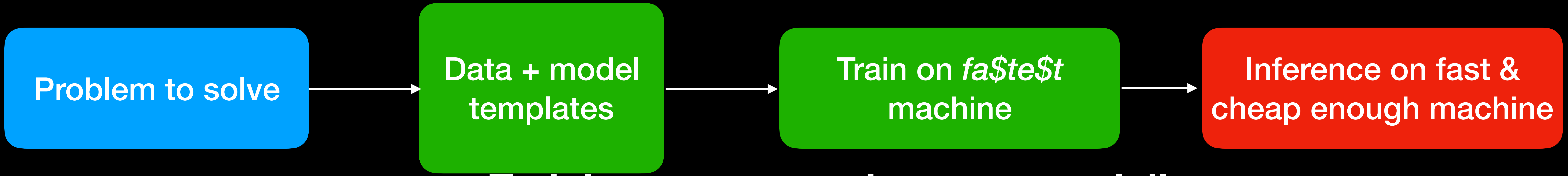
Problem to solve → Data + model templates → Train on *fa$te$t* machine → Inference on fast & cheap enough machine

Model size and compute cost growing fast



by Eugenio Culurciello

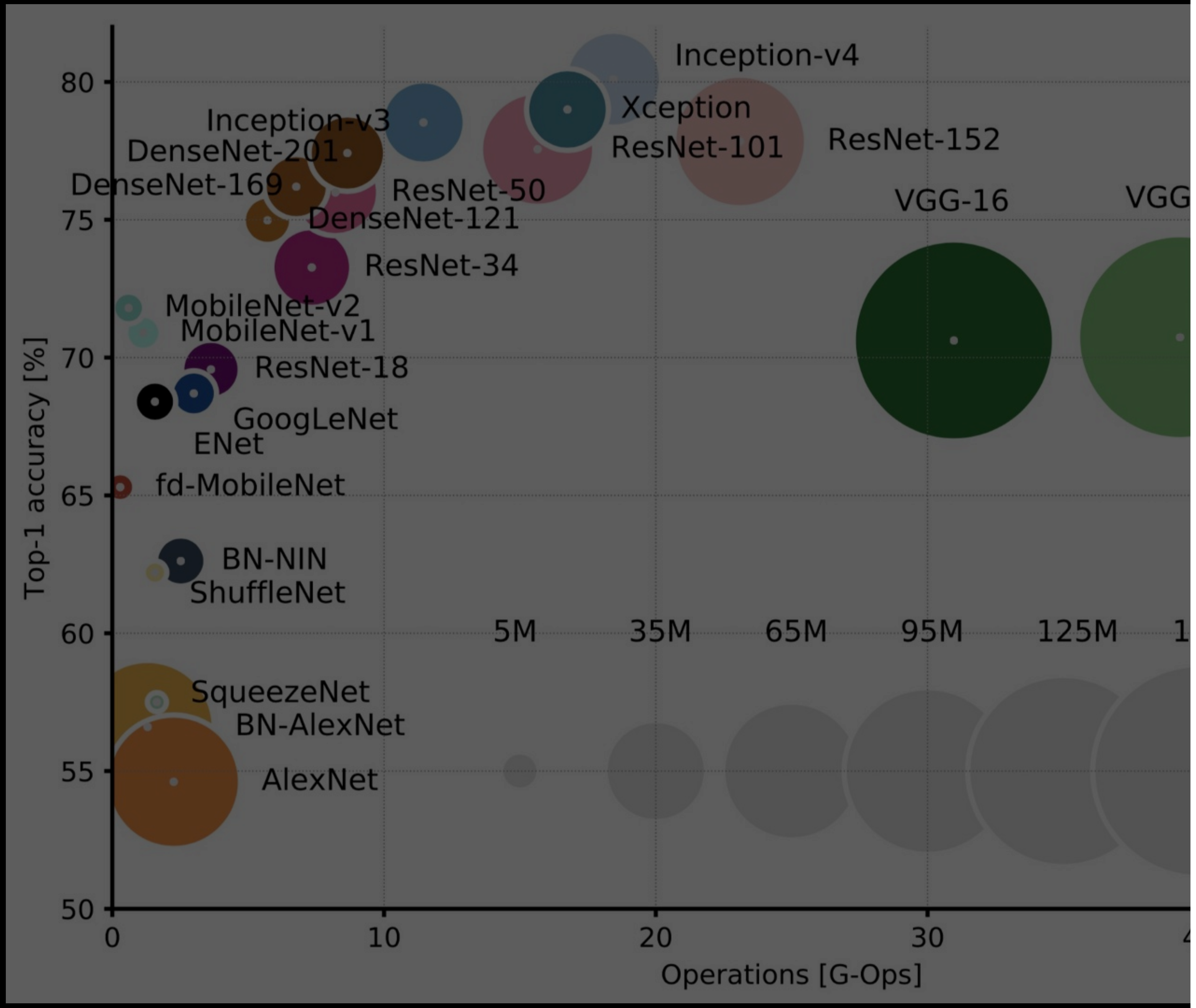
Machine learning era:



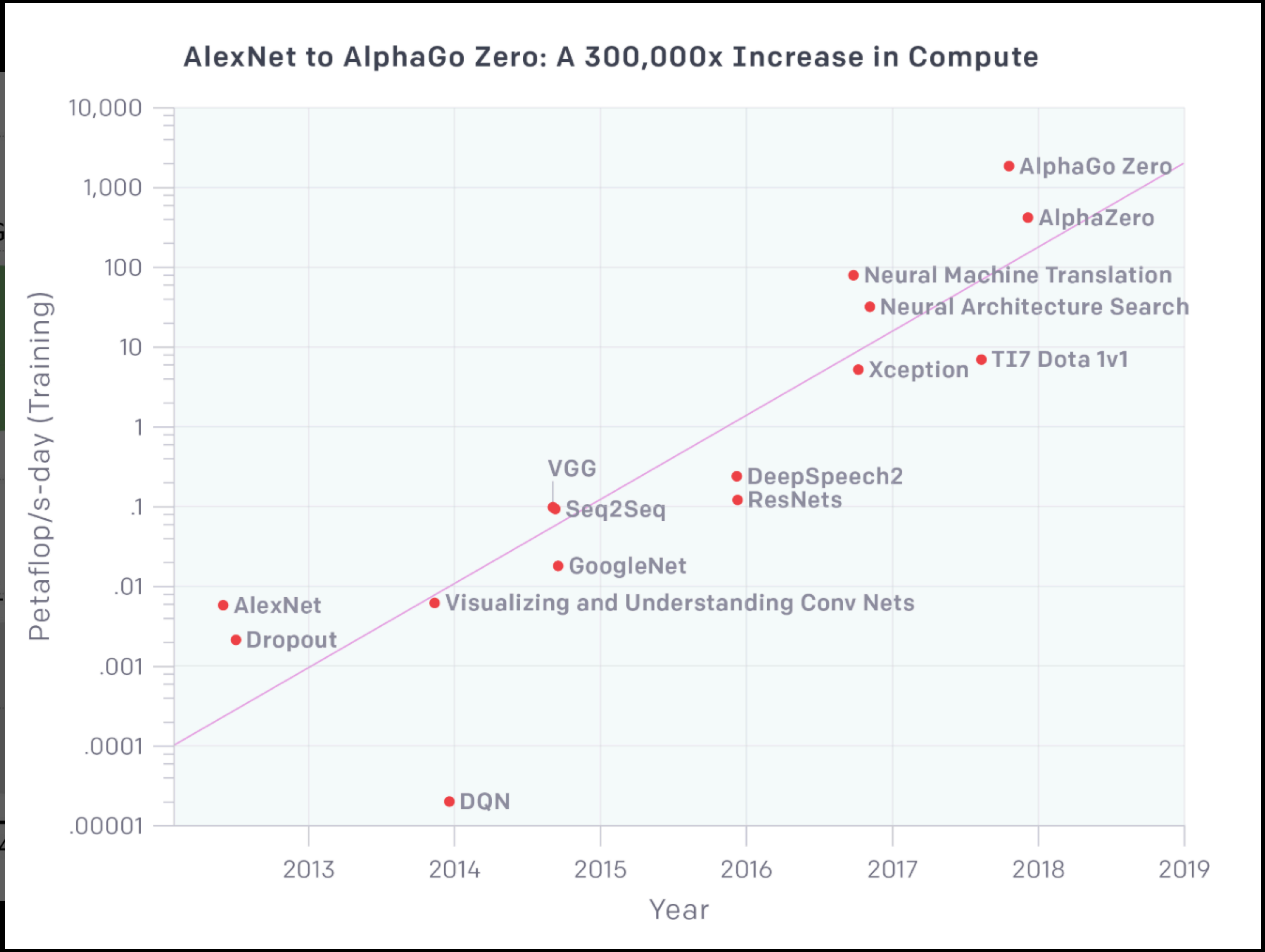
Problem to solve → Data + model templates → Train on *fa$te$t* machine → Inference on fast & cheap enough machine

Training costs growing exponentially

Model size and compute cost growing fast



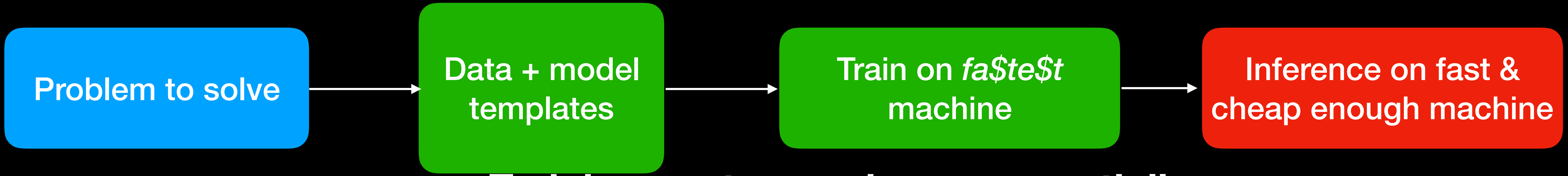
by Eugenio Culurciello



AlexNet to AlphaGo Zero: A 300,000x Increase in Compute

by Open AI

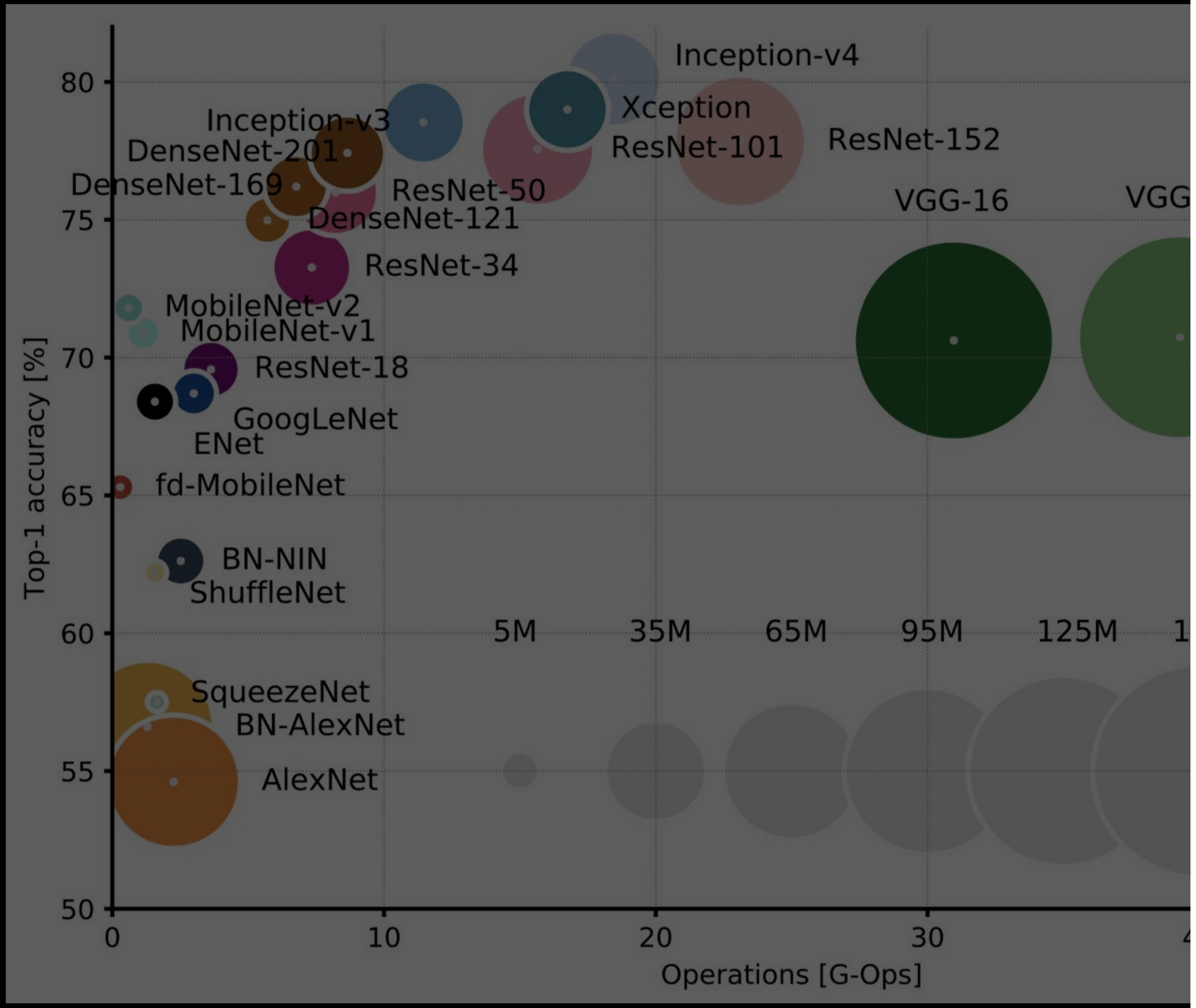
Machine learning era:



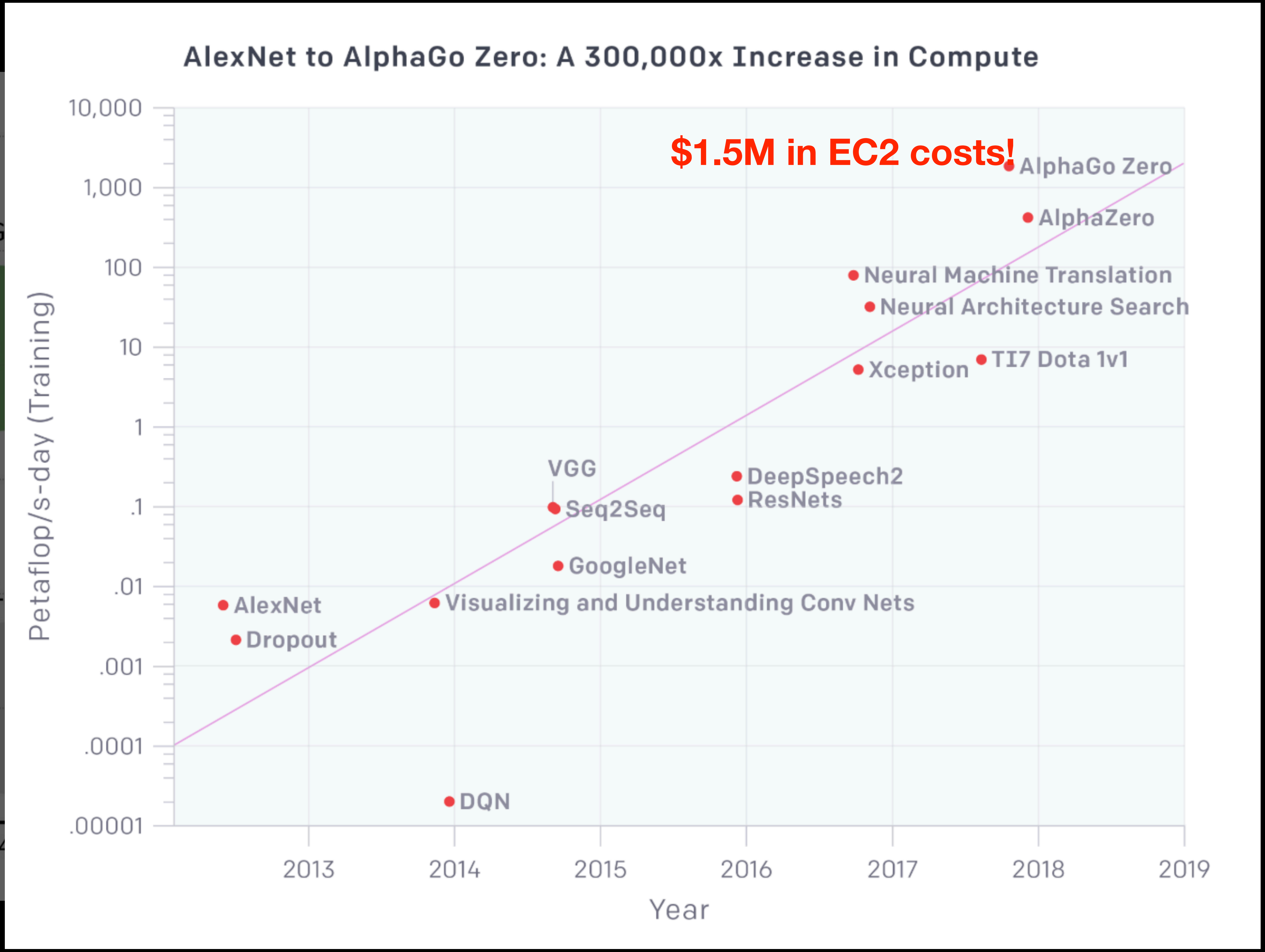
Problem to solve → Data + model templates → Train on *fa$te$t* machine → Inference on fast & cheap enough machine

Training costs growing exponentially

Model size and compute cost growing fast



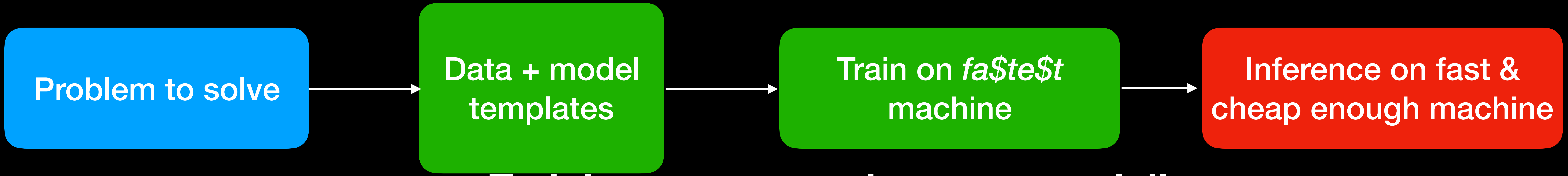
by Eugenio Culurciello



$1.5M in EC2 costs!

by Open AI

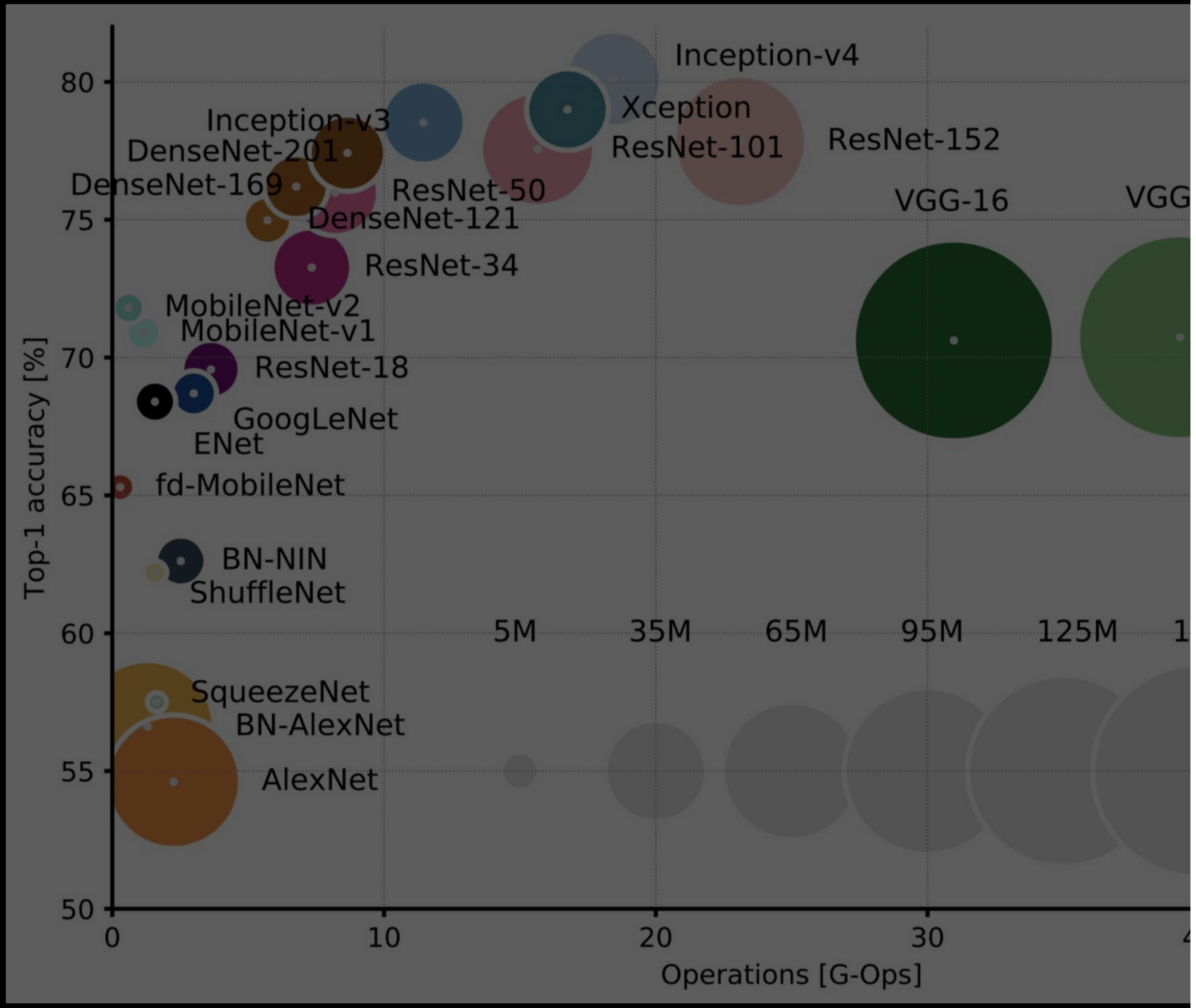
Machine learning era:



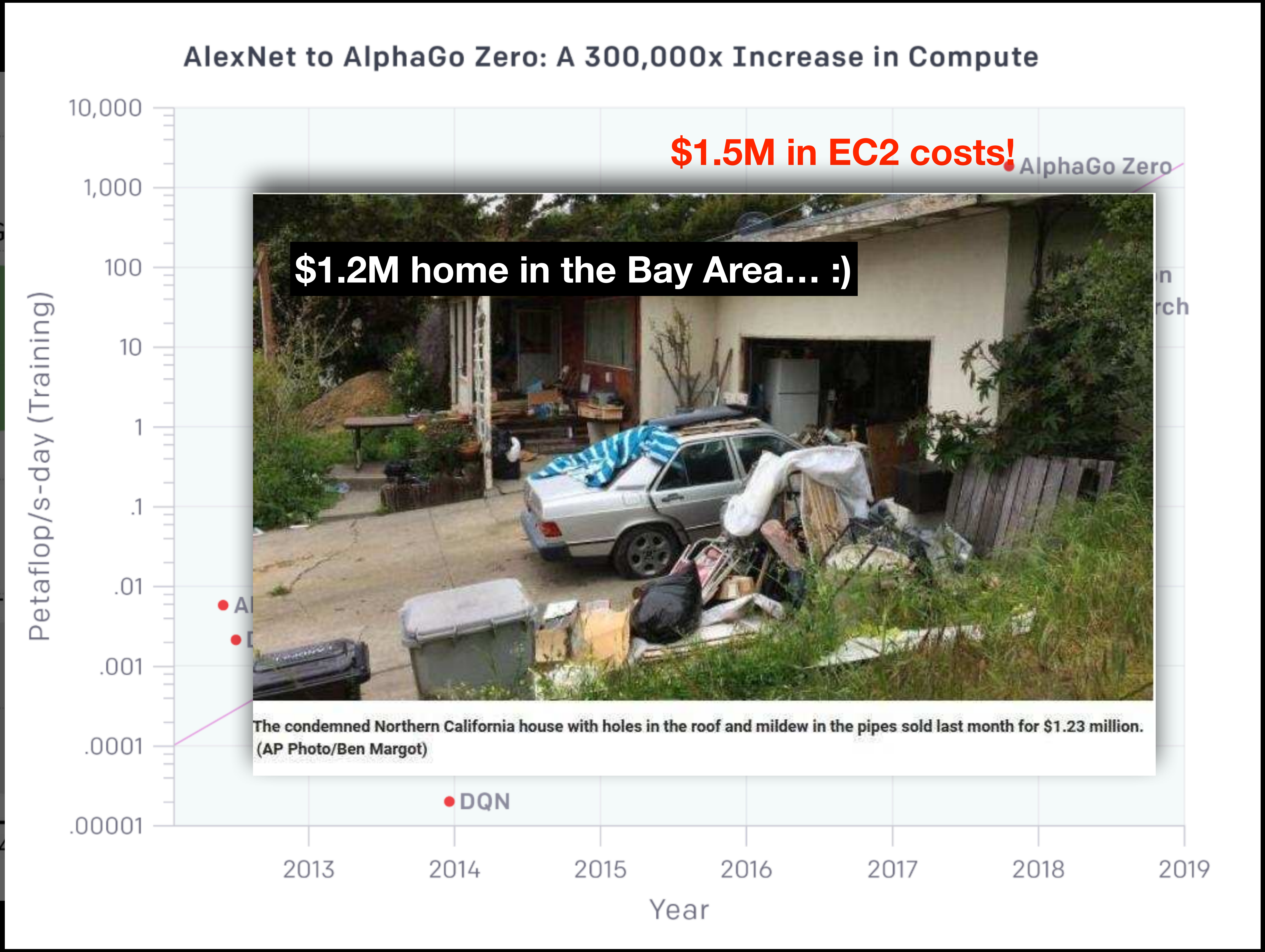
Problem to solve → Data + model templates → Train on *fa$te$t* machine → Inference on fast & cheap enough machine

Training costs growing exponentially

Model size and compute cost growing fast



by Eugenio Culurciello



$1.5M in EC2 costs!

$1.2M home in the Bay Area... :)

The condemned Northern California house with holes in the roof and mildew in the pipes sold last month for $1.23 million. (AP Photo/Ben Margot)

by Open AI

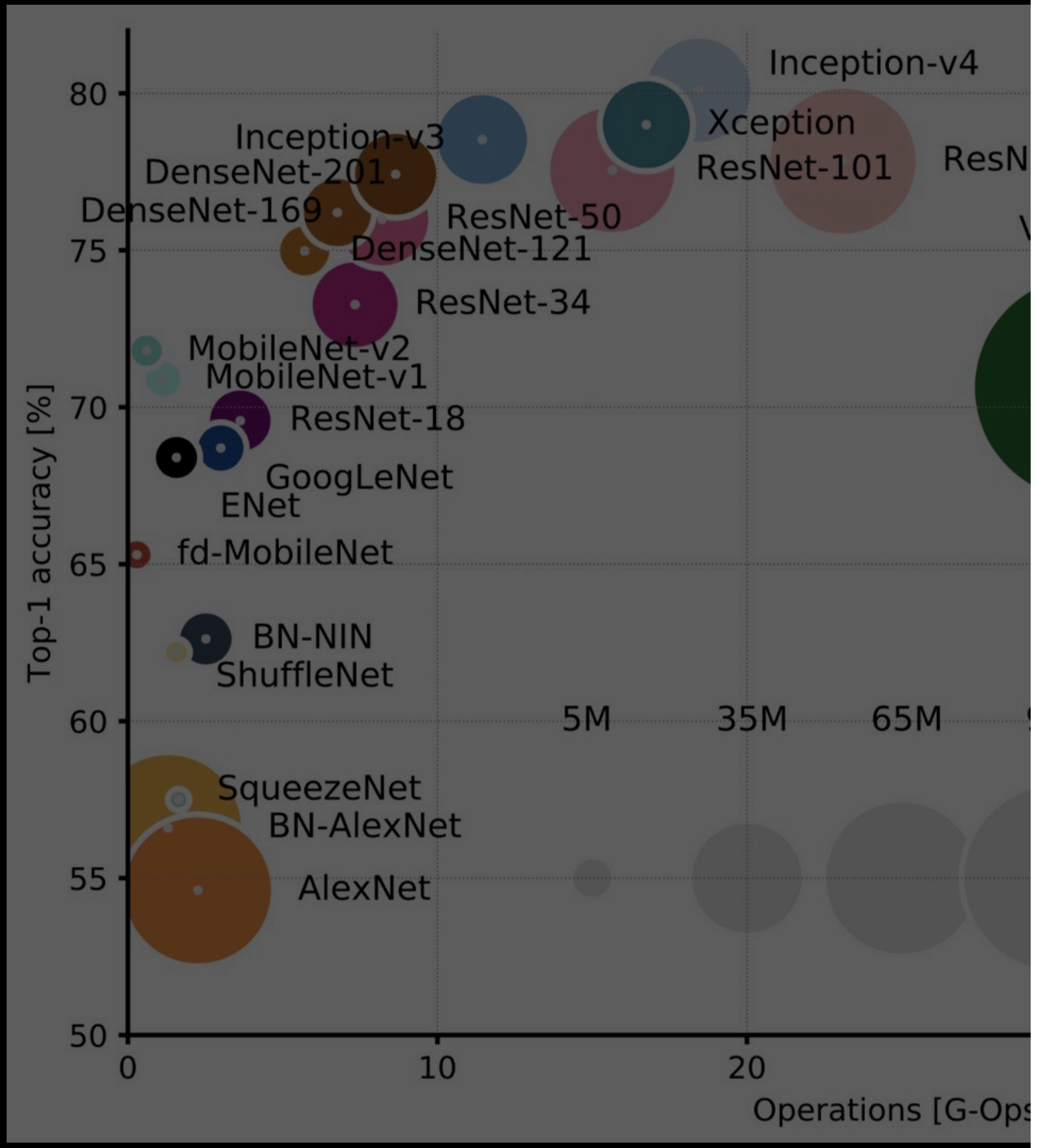
Machine learning era:

Problem to solv

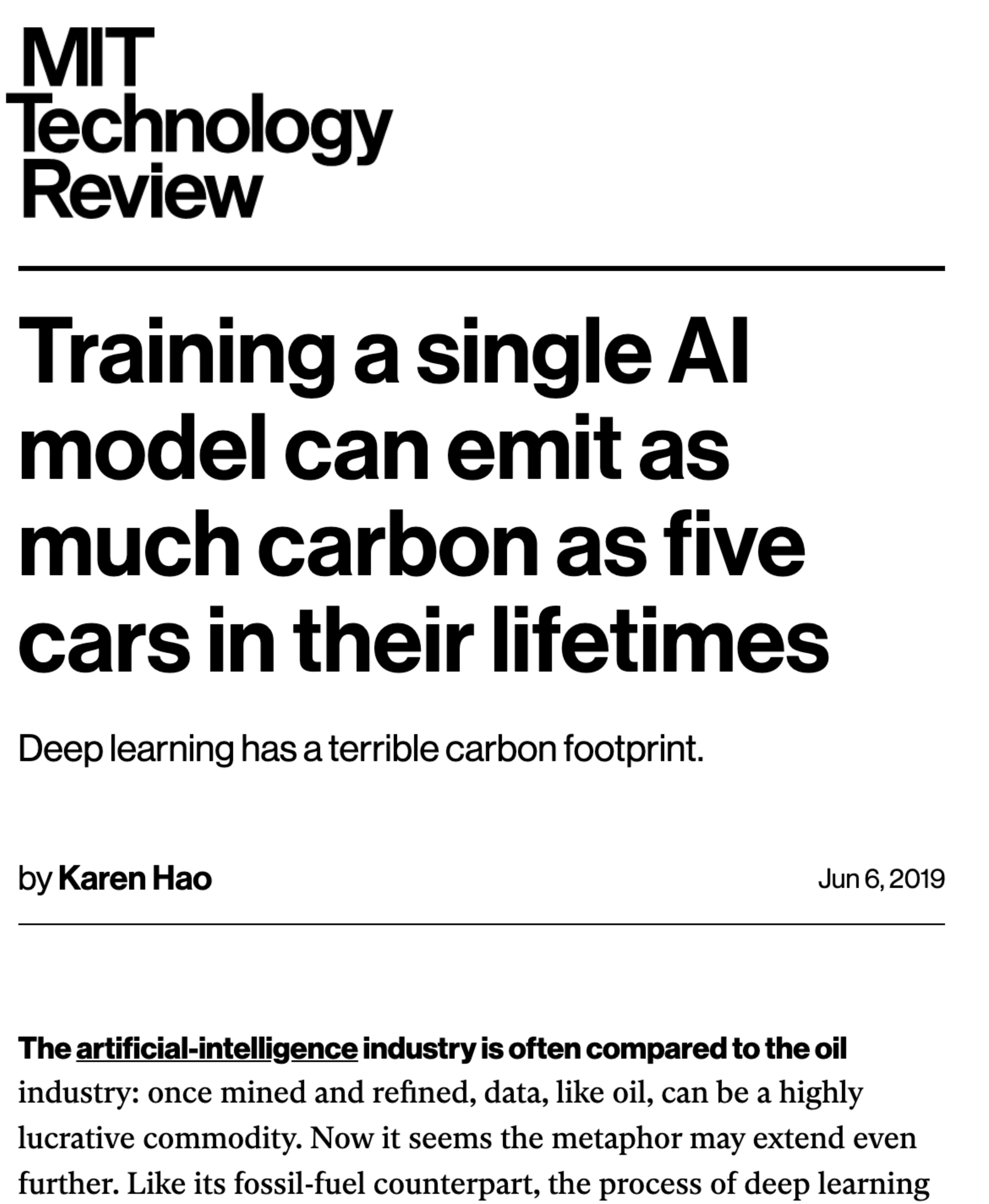
→ Inference on fast & cheap enough machine

ntially

Model size and compute cost growi



by Eugenio Culurciello



MIT Technology Review

Training a single AI model can emit as much carbon as five cars in their lifetimes

Deep learning has a terrible carbon footprint.

by **Karen Hao**

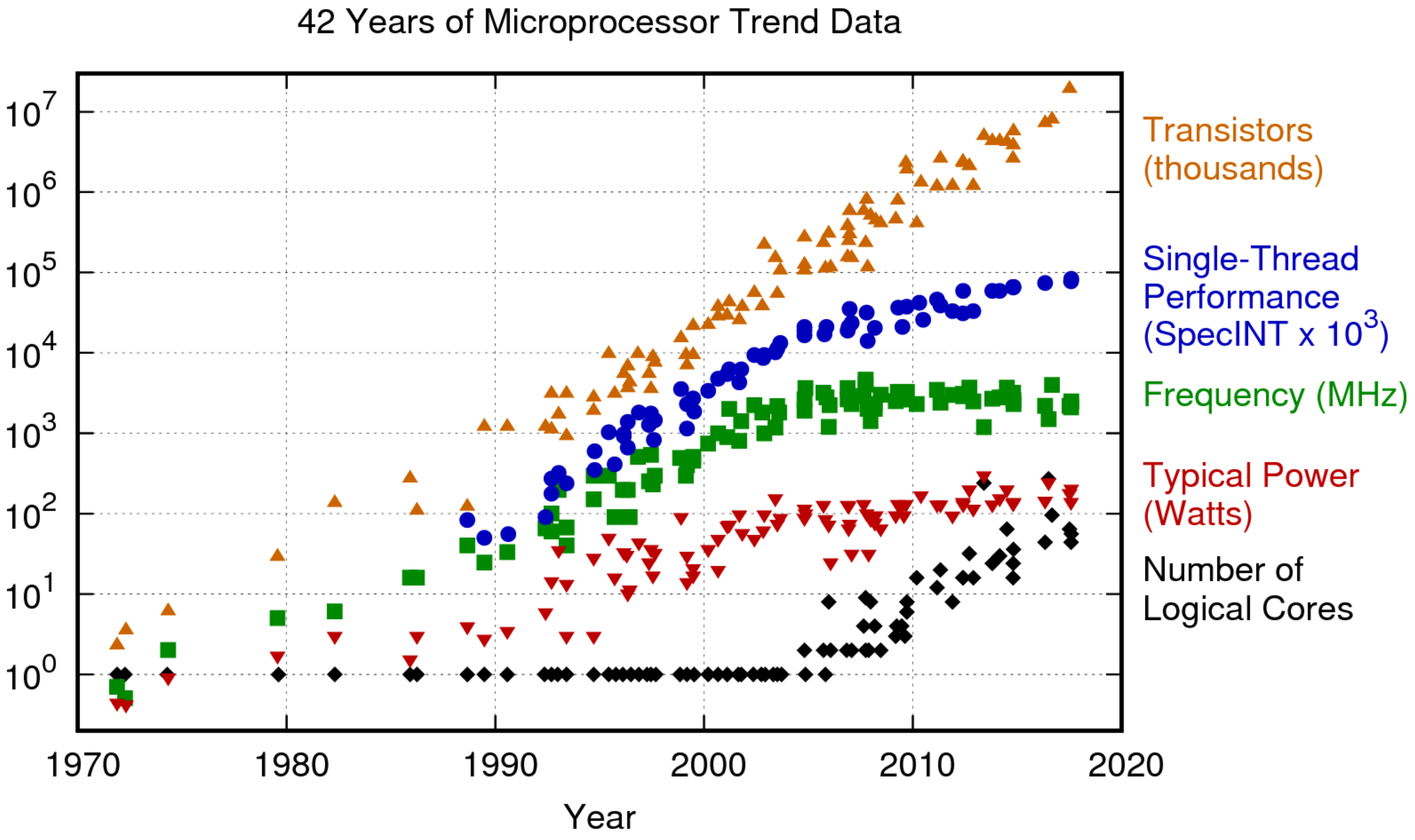
Jun 6, 2019

The artificial-intelligence industry is often compared to the oil industry: once mined and refined, data, like oil, can be a highly lucrative commodity. Now it seems the metaphor may extend even further. Like its fossil-fuel counterpart, the process of deep learning

nd mildew in the pipes sold last month for $1.23 million.

by Open AI

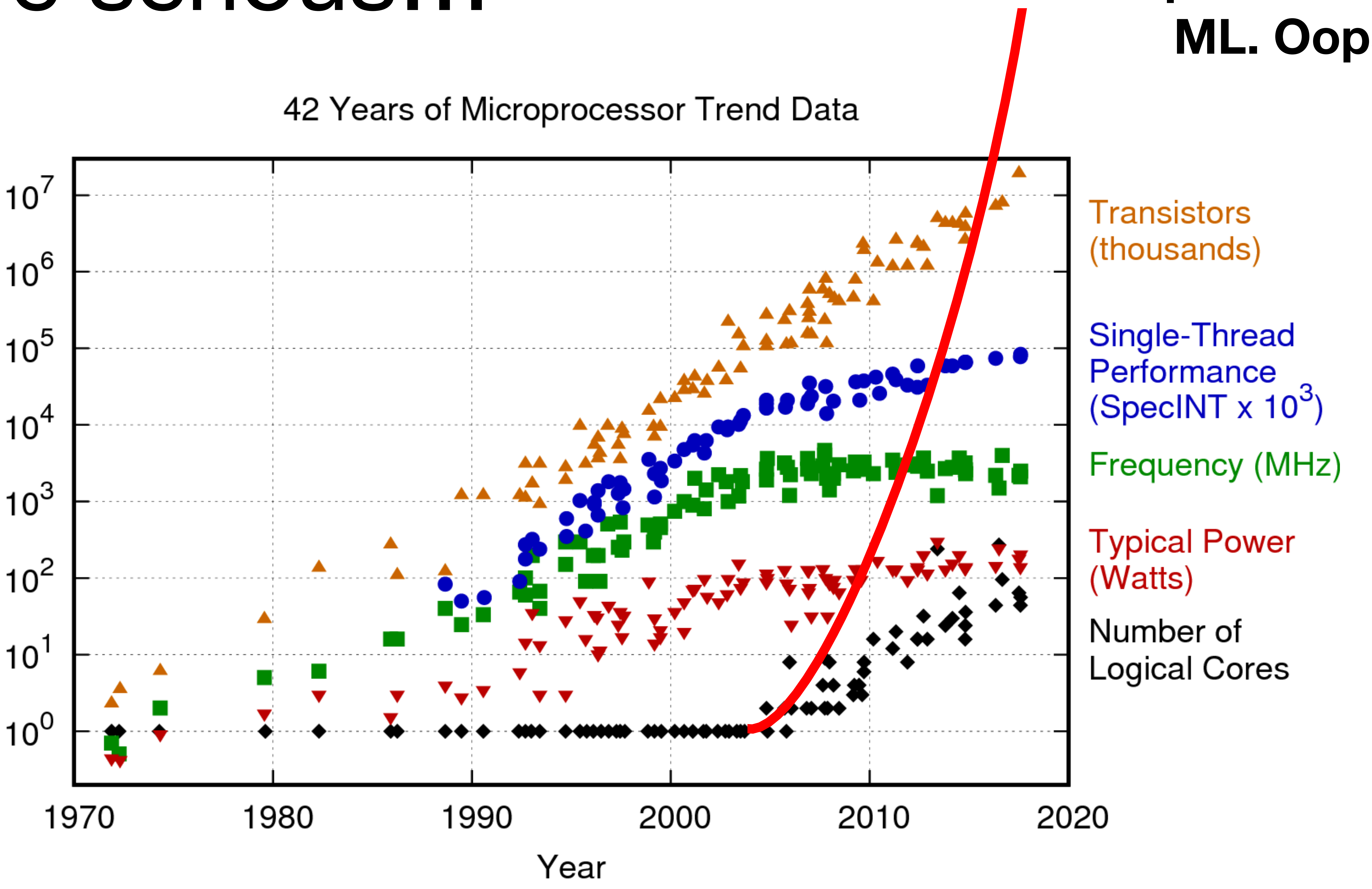
It gets more serious…



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

It gets more serious…

Computational cost of ML. Oops. :)



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

Impact of ML will be limited if we don’t squeeze as much efficiency as we can!

Impact of ML will be limited if we don’t squeeze as much efficiency as we can!

Model, SW and HW optimization are key…

A perfect storm

A perfect storm

Cambrian explosion of models, workloads, and use cases.

CNN GAN RNN MLP DQNN

A perfect storm

Growing set of requirements: **cost, latency, power, security & privacy**

Cambrian explosion of models, workloads, and use cases.	CNN	GAN	RNN	MLP	DQNN

A perfect storm

Growing set of requirements: **cost, latency, power, security & privacy**

Cambrian explosion of models, workloads, and use cases.

CNN GAN RNN MLP DQNN

Silicon scaling limitations (Dennard and Moore):

Cambrian explosion of HW backends. Heterogeneous HW.

intel NVIDIA XILINX Microsoft QUALCOMM amazon Google HUAWEI

A perfect storm

Growing set of requirements: **cost, latency, power, security & privacy**

Cambrian explosion of models, workloads, and use cases.

CNN GAN RNN MLP DQNN

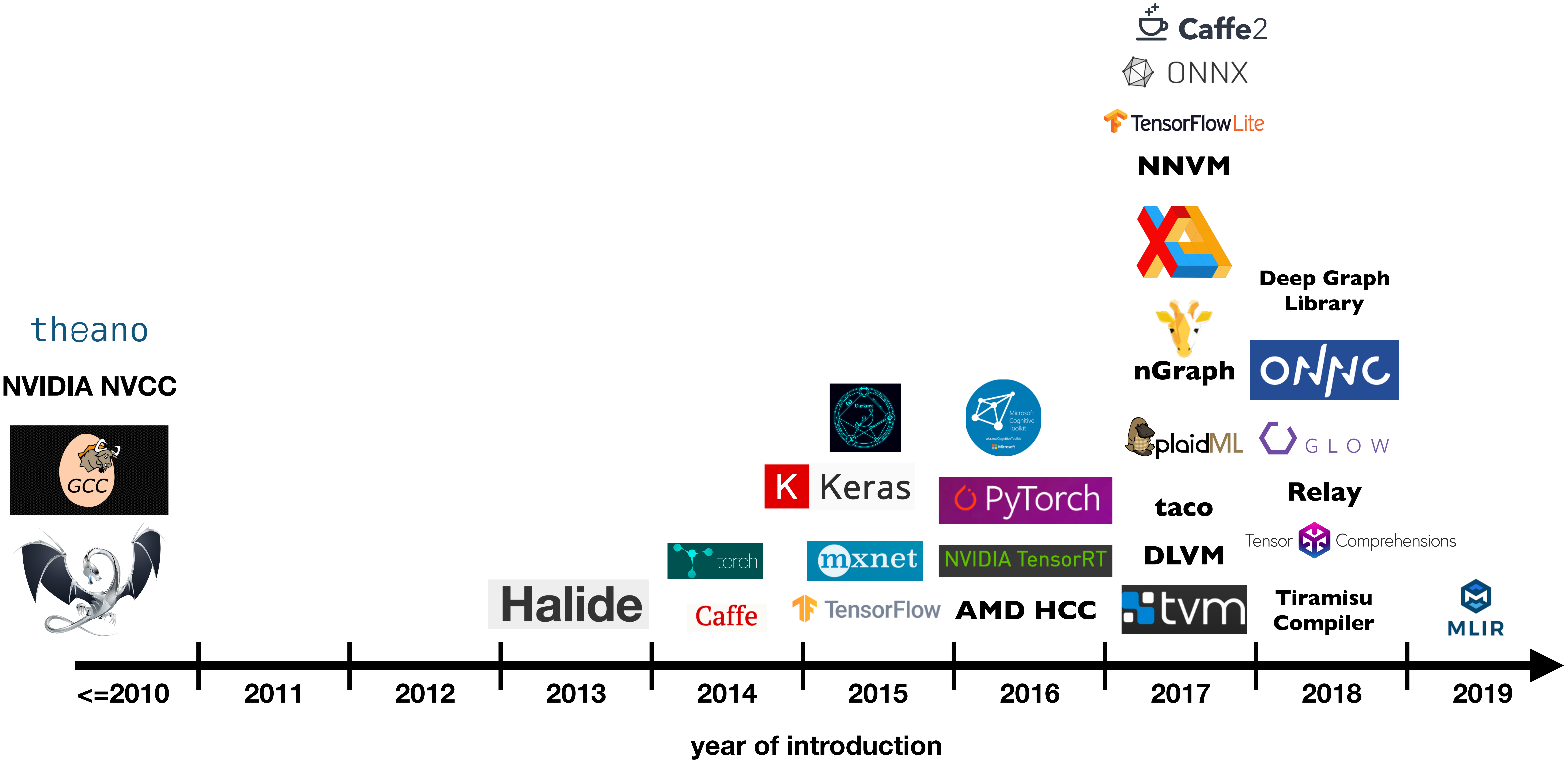
Rapidly evolving ML software ecosystem quickly fragmenting

Silicon scaling limitations (Dennard and Moore):

Cambrian explosion of HW backends. Heterogeneous HW.

intel NVIDIA XILINX Microsoft QUALCOMM amazon Google HUAWEI

Deep learning “stack” (r?)evolution



<=2010: theano, NVIDIA NVCC, GCC

2013: Halide

2014: torch, Caffe

2015: Darknet, Keras, mxnet, TensorFlow

2016: Microsoft Cognitive Toolkit, PyTorch, NVIDIA TensorRT, AMD HCC

2017: Caffe2, ONNX, TensorFlowLite, NNVM, nGraph, plaidML, taco, DLVM, tvm

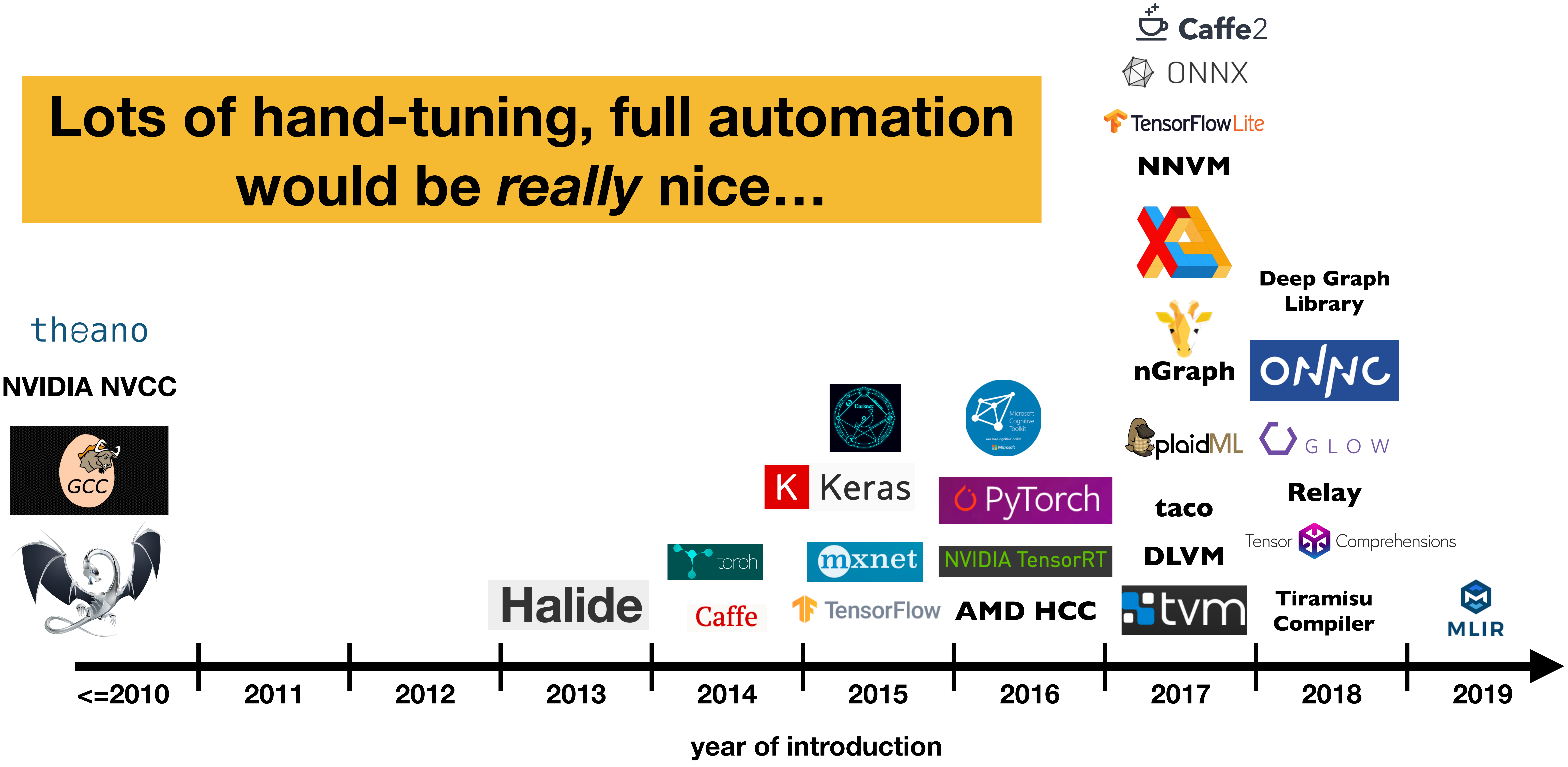
2018: Deep Graph Library, ONNC, GLOW, Relay, Tensor Comprehensions, Tiramisu Compiler

2019: MLIR

year of introduction

Deep learning "stack" (r?)evolution

Lots of hand-tuning, full automation would be *really* nice…



<=2010: theano, NVIDIA NVCC, GCC

2013: Halide

2014: torch, Caffe

2015: Keras, mxnet, TensorFlow

2016: PyTorch, NVIDIA TensorRT, AMD HCC

2017: Caffe2, ONNX, TensorFlowLite, NNVM, nGraph, plaidML, taco, DLVM, tvm

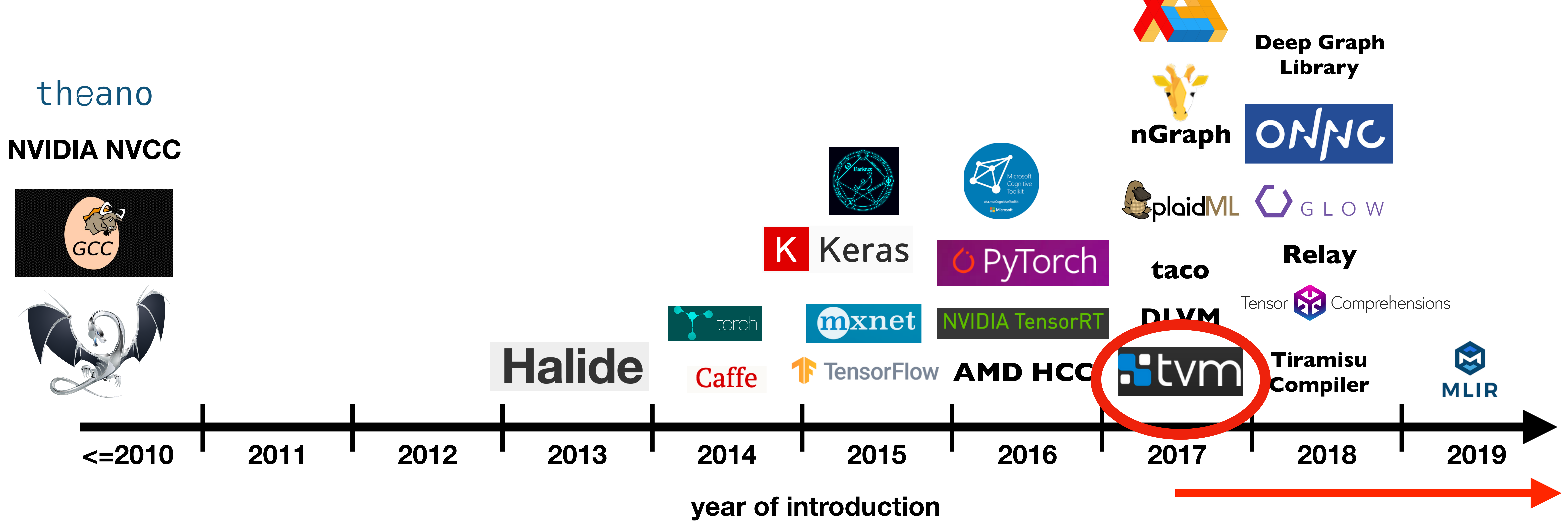
2018: Deep Graph Library, ONNC, GLOW, Relay, Tensor Comprehensions, Tiramisu Compiler

2019: MLIR

year of introduction

Deep learning “stack” (r?)evolution

Lots of hand-tuning, full automation would be *really* nice…



Caffe2

ONNX

TensorFlow Lite

NNVM

theano

NVIDIA NVCC

GCC

Halide

torch

Caffe

Keras

mxnet

TensorFlow

Microsoft Cognitive Toolkit

PyTorch

NVIDIA TensorRT

AMD HCC

nGraph

plaidML

taco

DLVM

tvm

Deep Graph Library

ONNC

GLOW

Relay

Tensor Comprehensions

Tiramisu Compiler

MLIR

<=2010

2011

2012

2013

2014

2015

2016

2017

2018

2019

year of introduction

Current Dominant Deep Learning Systems Landscape

Orchestrators

Kubeflow

SELDON

ALGORITHMIA

Azure ML

GCP Datalab

Frameworks and Inference engines

K

m

ONNX RUNTIME

TensorRT

DL Compilers

nGraph

GLOW

MLIR

Kernel Libraries

cuDNN

NNPack

MKL-DNN

Hand optimized

Hardware

Current Dominant Deep Learning Systems Landscape

Orchestrators

Kubeflow SELDON ALGORITHMIA Azure ML GCP Datalab

Frameworks and Inference engines

ONNX RUNTIME TensorRT

DL Compilers

nGraph GLOW MLIR

Kernel Libraries

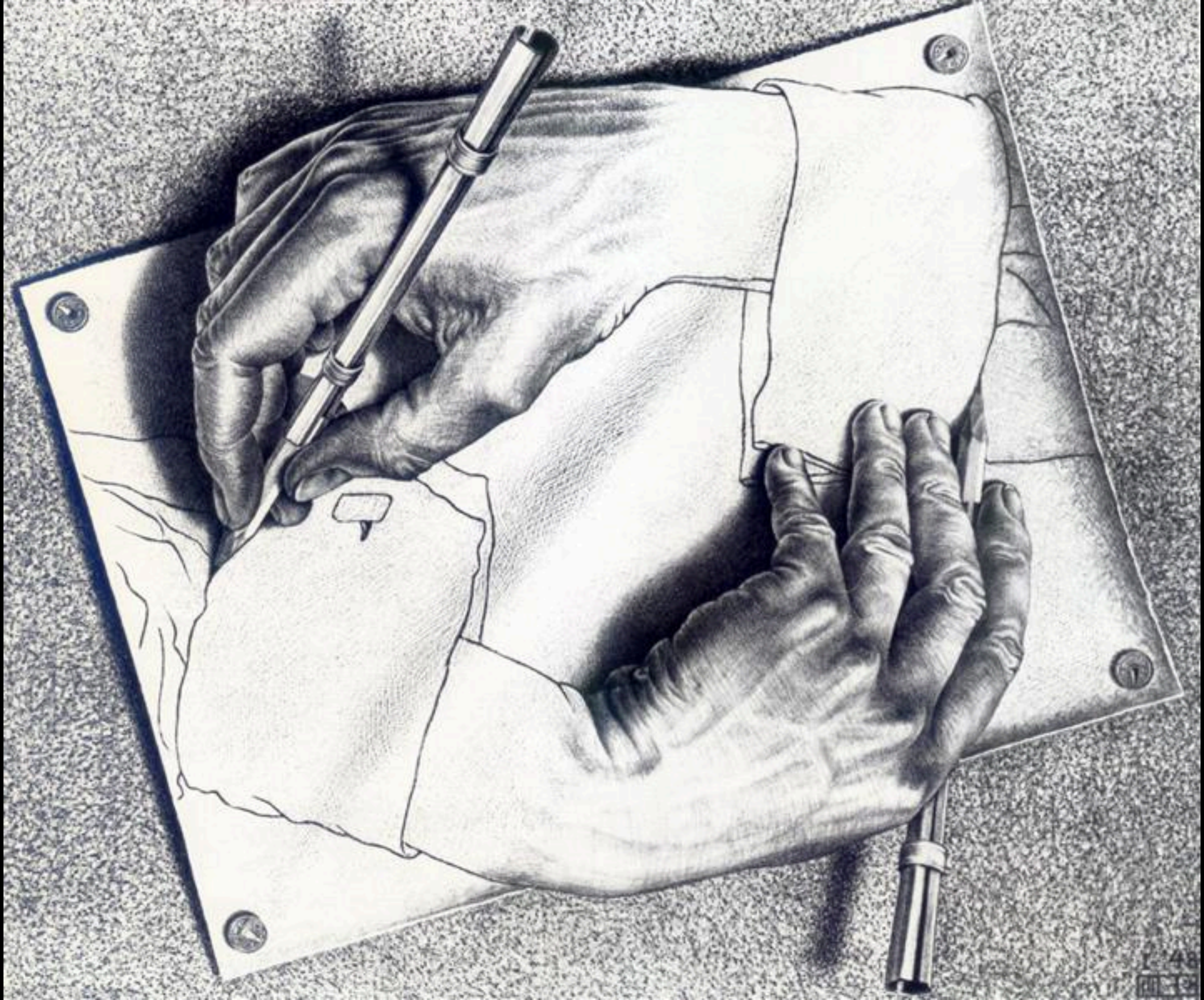
cuDNN NNPack MKL-DNN

Hand optimized

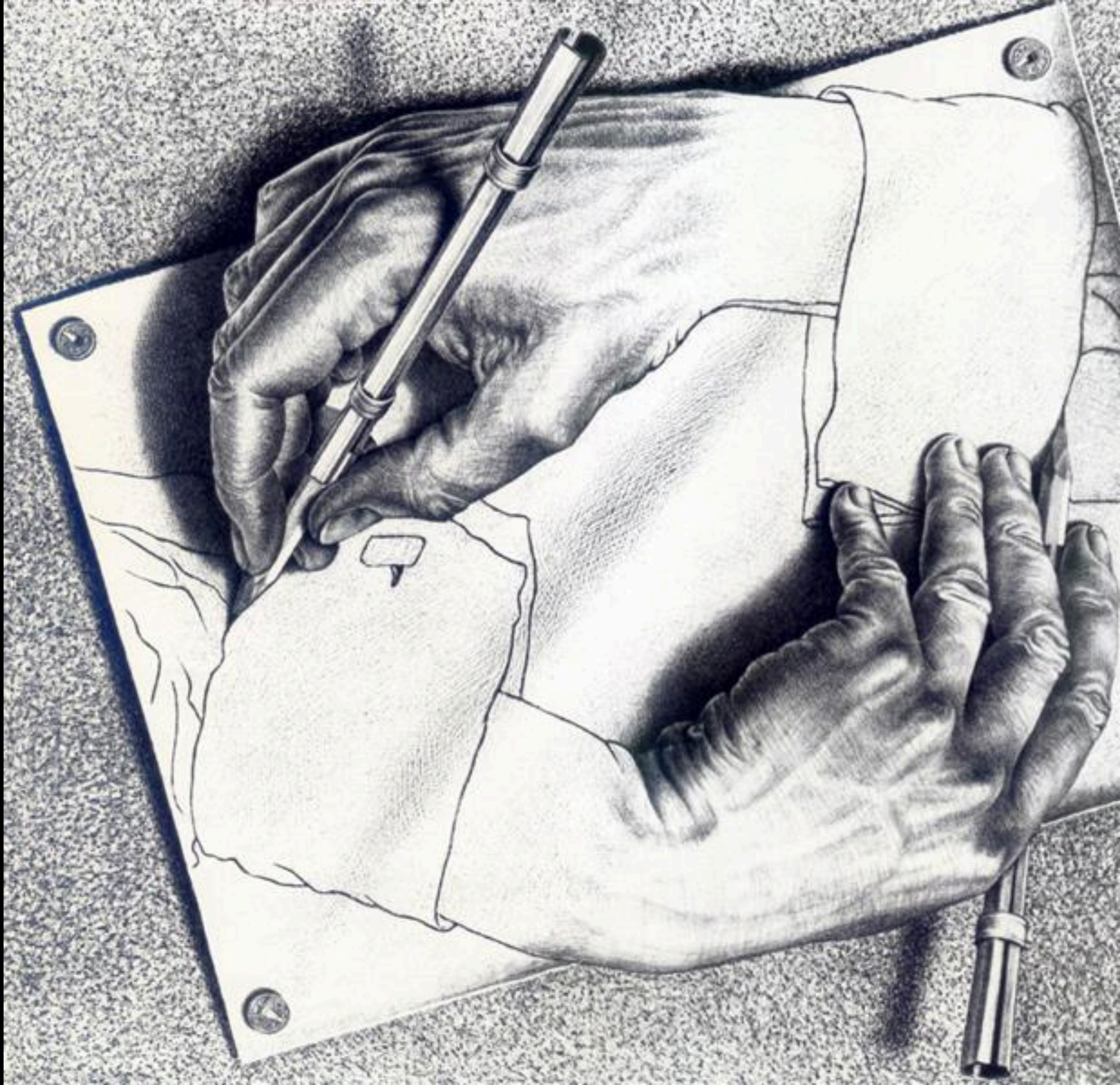
Hardware

tvm

Open source, **automated** end-to-end optimization framework for deep learning.

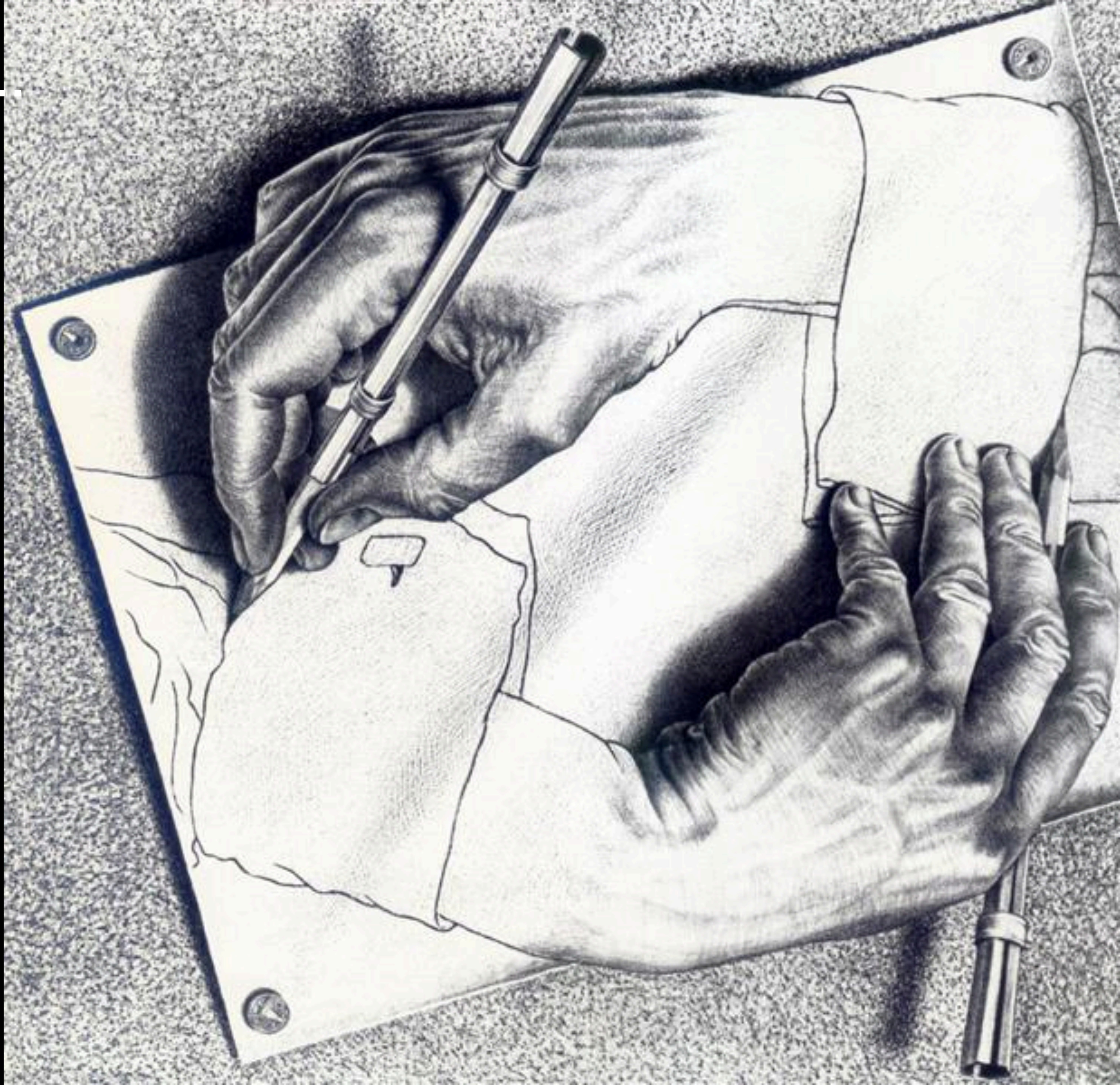


I '48



Using ML for better ML systems...

Deal with design complexity and large parameter spaces...



Using ML for better ML systems…

Deal with design complexity and large parameter spaces…

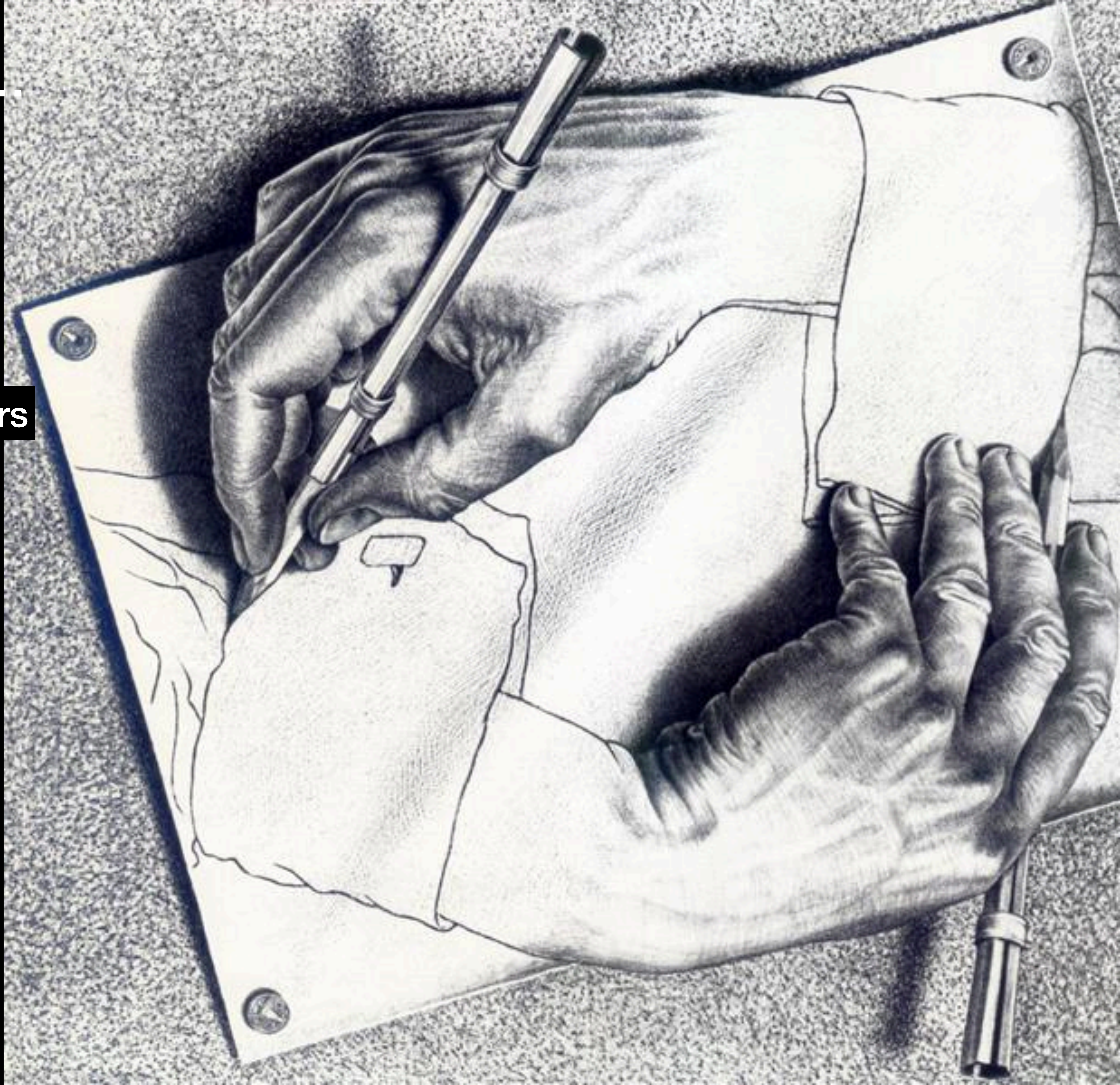
Model optimization strategies and parameters

Efficient operator implementations

Data communication patterns

Model-HW co-tuning

Searching for efficient HW designs



This past year…

This past year…

Broader model coverage (e.g., PyTorch integration, RelayVM, BERT, SSD)

This past year…

Broader model coverage (e.g., PyTorch integration, RelayVM, BERT, SSD)

More hardware backends (e.g., CortexM, RISC-V, DSPs)

This past year…

Broader model coverage (e.g., PyTorch integration, RelayVM, BERT, SSD)

More optimizations (e.g., quantization, data layout)

More hardware backends (e.g., CortexM, RISC-V, DSPs)

This past year…

Broader model coverage (e.g., PyTorch integration, RelayVM, BERT, SSD)

More optimizations (e.g., quantization, data layout)

More hardware backends (e.g., CortexM, RISC-V, DSPs)

Usability (tutorials, docs, automation), community development

Open Source Community Growth and Impact

70% growth from Dec 2018 to **295 contributors** from UW, Berkeley, Cornell, UCLA, Amazon, Huawei, NTT, Facebook, Microsoft, Qualcomm, Alibaba, Intel, …

tvm Open Source Community Growth and Impact

70% growth from Dec 2018 to **295 contributors** from UW, Berkeley, Cornell, UCLA, Amazon, Huawei, NTT, Facebook, Microsoft, Qualcomm, Alibaba, Intel, …

Used in production at leading vendors:

aws

Deep Learning Compiler Service

HUAWEI

Tensor Engine for mobile ASIC

f

Mobile and Server Optimizations

Microsoft

Cloud-side model optimization

tvm Open Source Community Growth and Impact

70% growth from Dec 2018 to **295 contributors** from UW, Berkeley, Cornell, UCLA, Amazon, Huawei, NTT, Facebook, Microsoft, Qualcomm, Alibaba, Intel, …

Used in production at leading vendors:

aws

Deep Learning Compiler Service

HUAWEI

Tensor Engine for mobile ASIC

f

Mobile and Server Optimizations

Microsoft

Cloud-side model optimization

THE APACHE SOFTWARE FOUNDATION

Incubated as Apache TVM recently. Independent governance, allowing competitors to collaborate.

tvm Open Source Community Growth and Impact

70% growth om Dec 20 to **295 co** **ibutors** n UW, Be eley, Cor , UCLA, Amazon, Huawei, NTT, Facebook, Microsoft, Qualcomm, Alibaba, Intel, …

Used in production at leading vendors:

aws

Deep Learning
Comp ervice

HUAWEI

Tensor Engine
r mobile

Mobile and Server
imizatio

osoft

Cloud-side model
imization

THE APACHE SOFTWARE FOUNDATION

Incubated as Apache TVM recently. Independent governance, allowing competitors to collaborate.

Jeff Gehlhaar

QUALCOMM®

Dec 2019

University of Washington

@qualcomm

Qualcomm

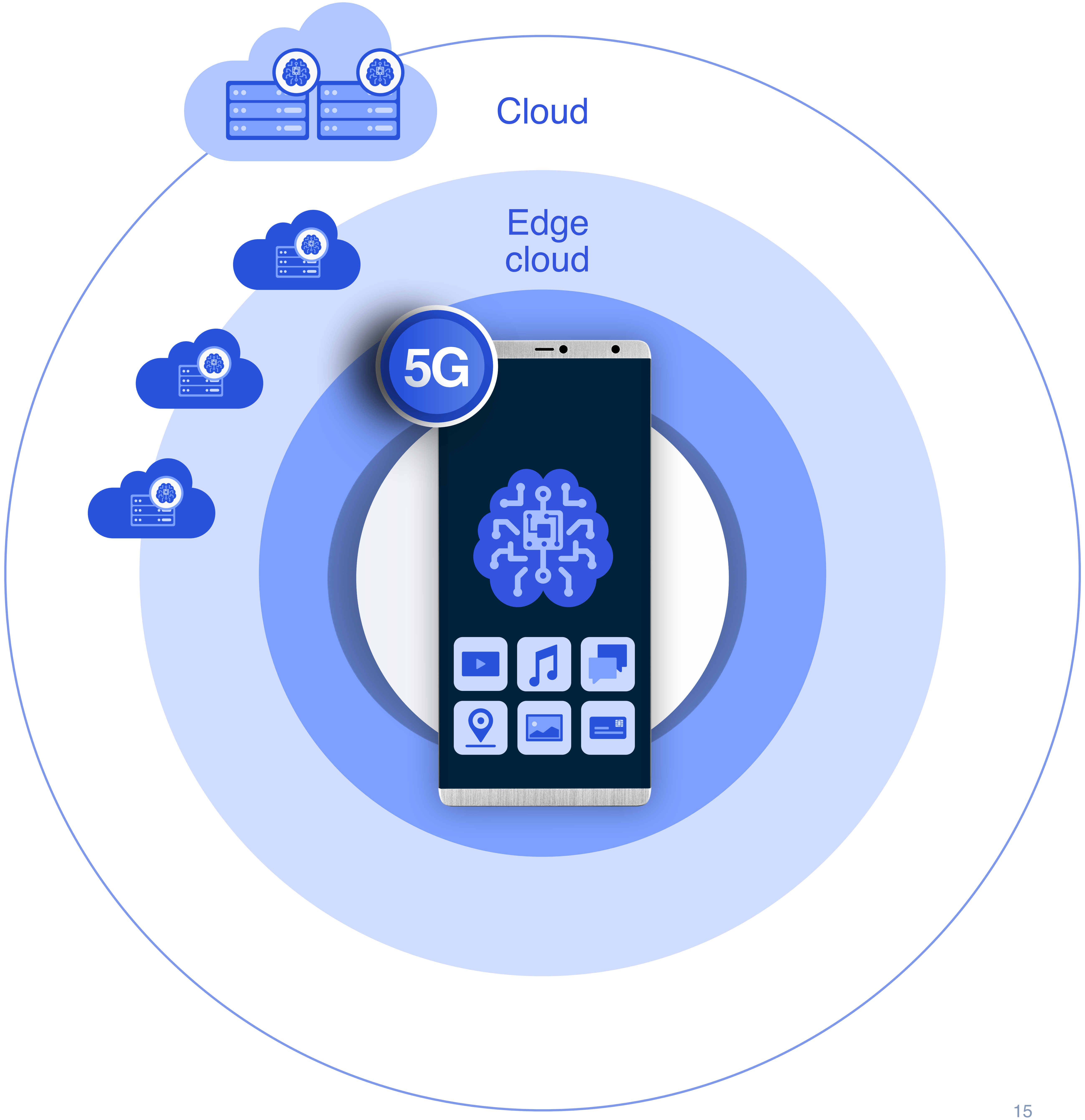
Qualcomm Technologies, Inc. AI Overview

Jeff Gehlhaar, VP Technology

Qualcomm Technologies, Inc.

We’re creating a future of distributed intelligence

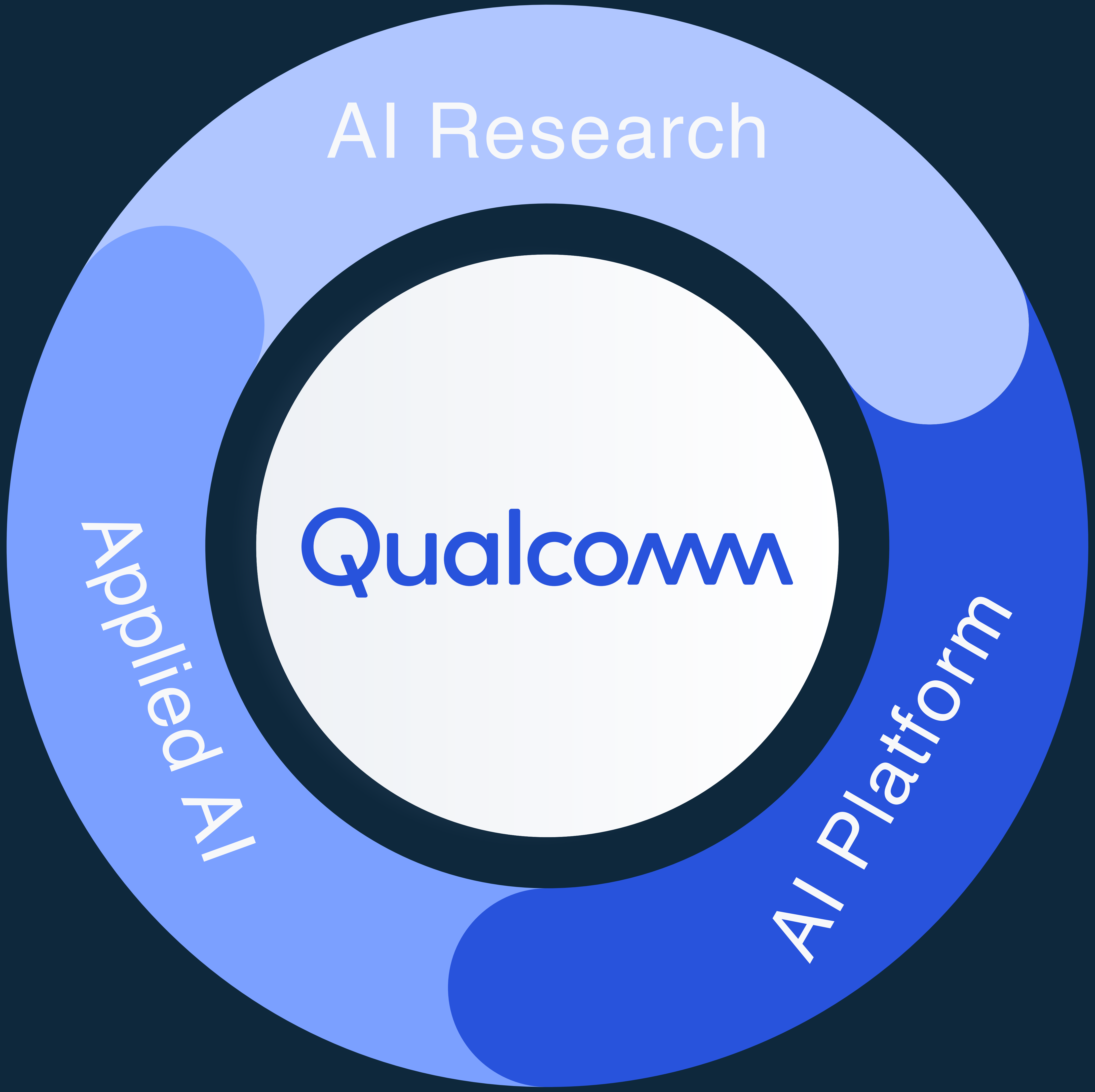
Our platforms are enabling a world of decentralized computing to realize the true potential of AI at scale. On-device inference processes data closest to the source for maximum speed and security, and low-latency 5G connectivity augments experiences with edge cloud processing for training updates and connected services.



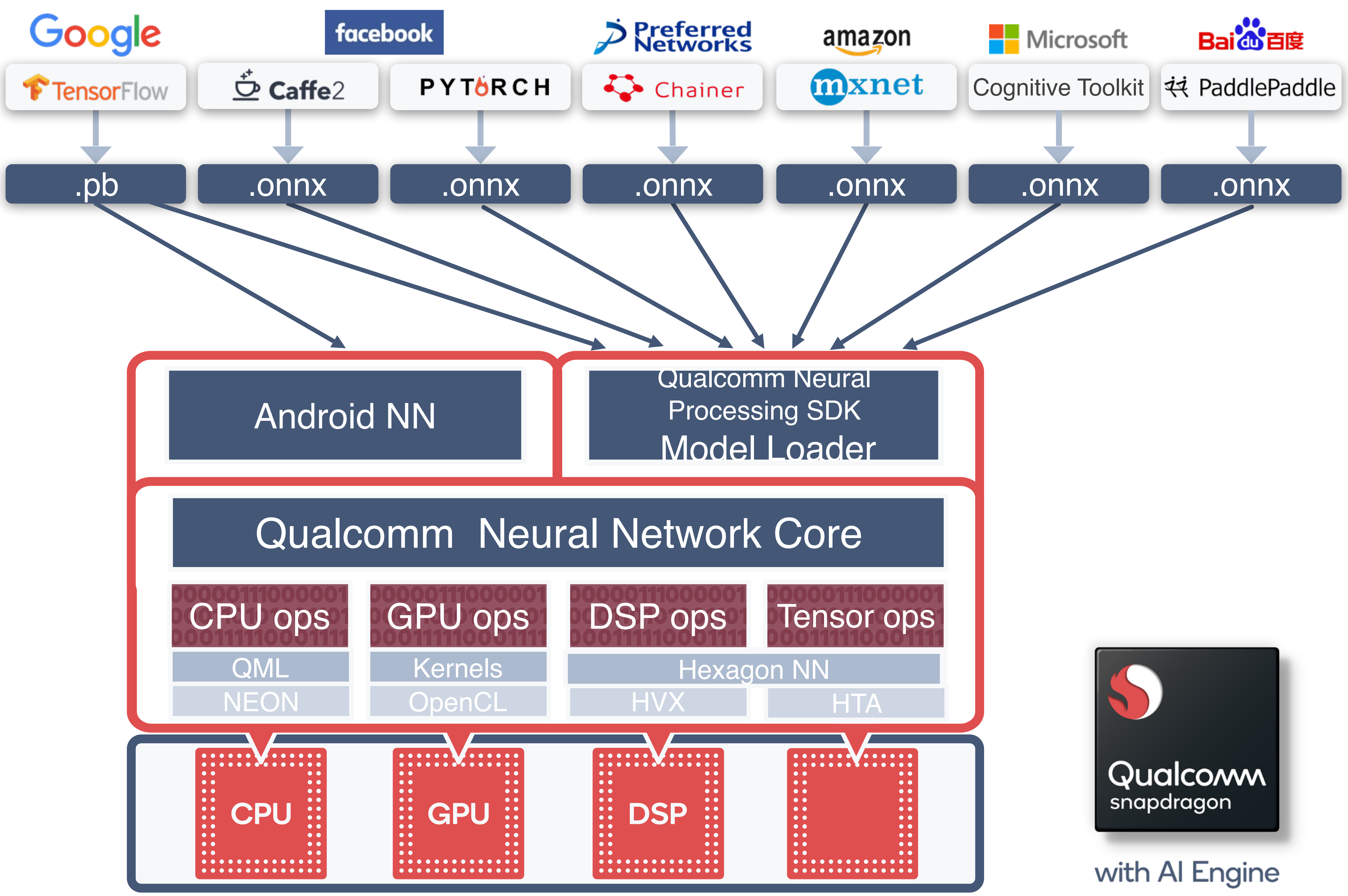
Our process

We design and develop holistic AI systems

Our process provides a comprehensive approach to AI research and development. We take on hard problems and tackle complexity head on to meticulously design and build systems that deliver complete end-to-end AI solutions, from fundamental research to product execution.

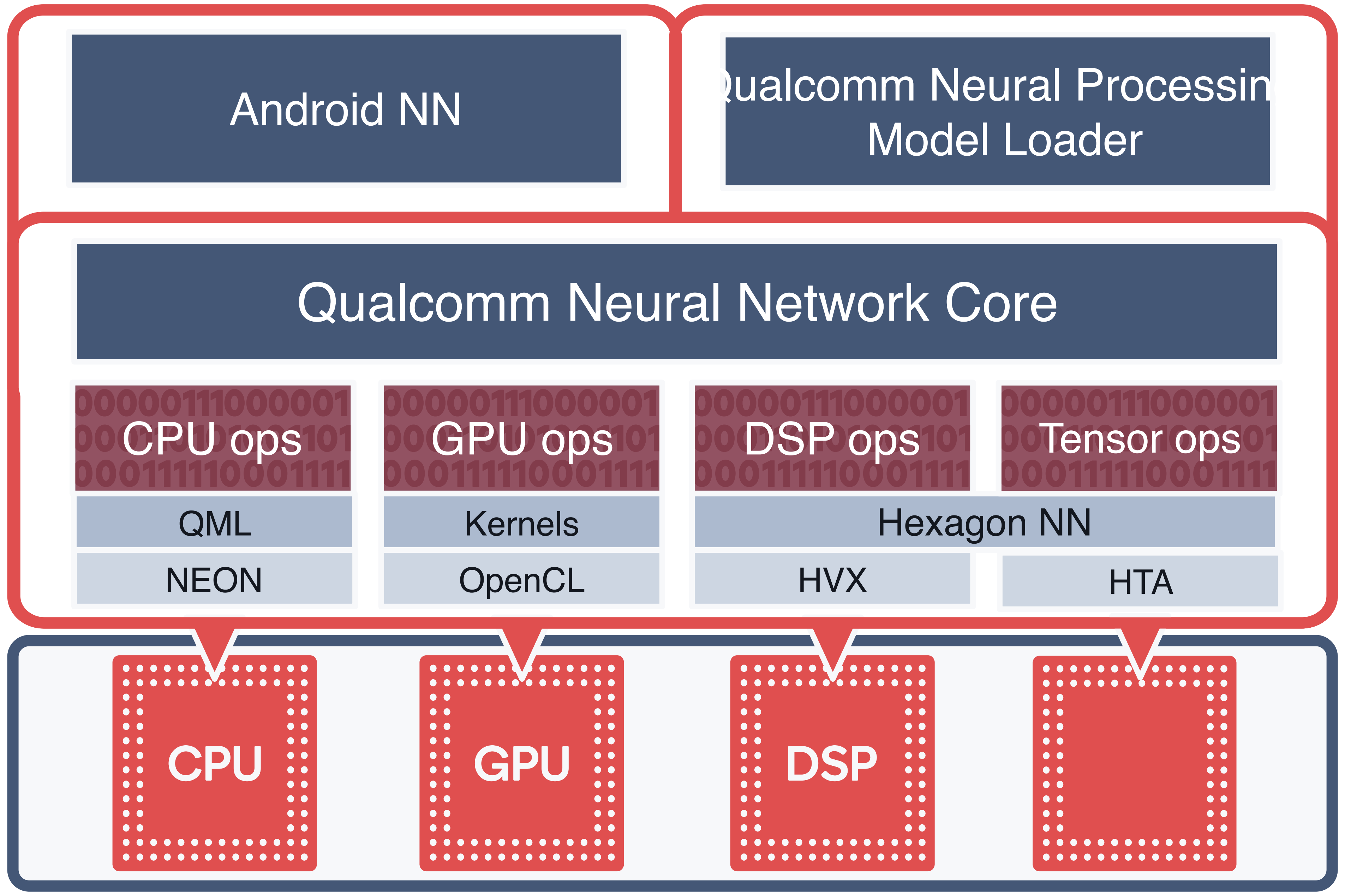


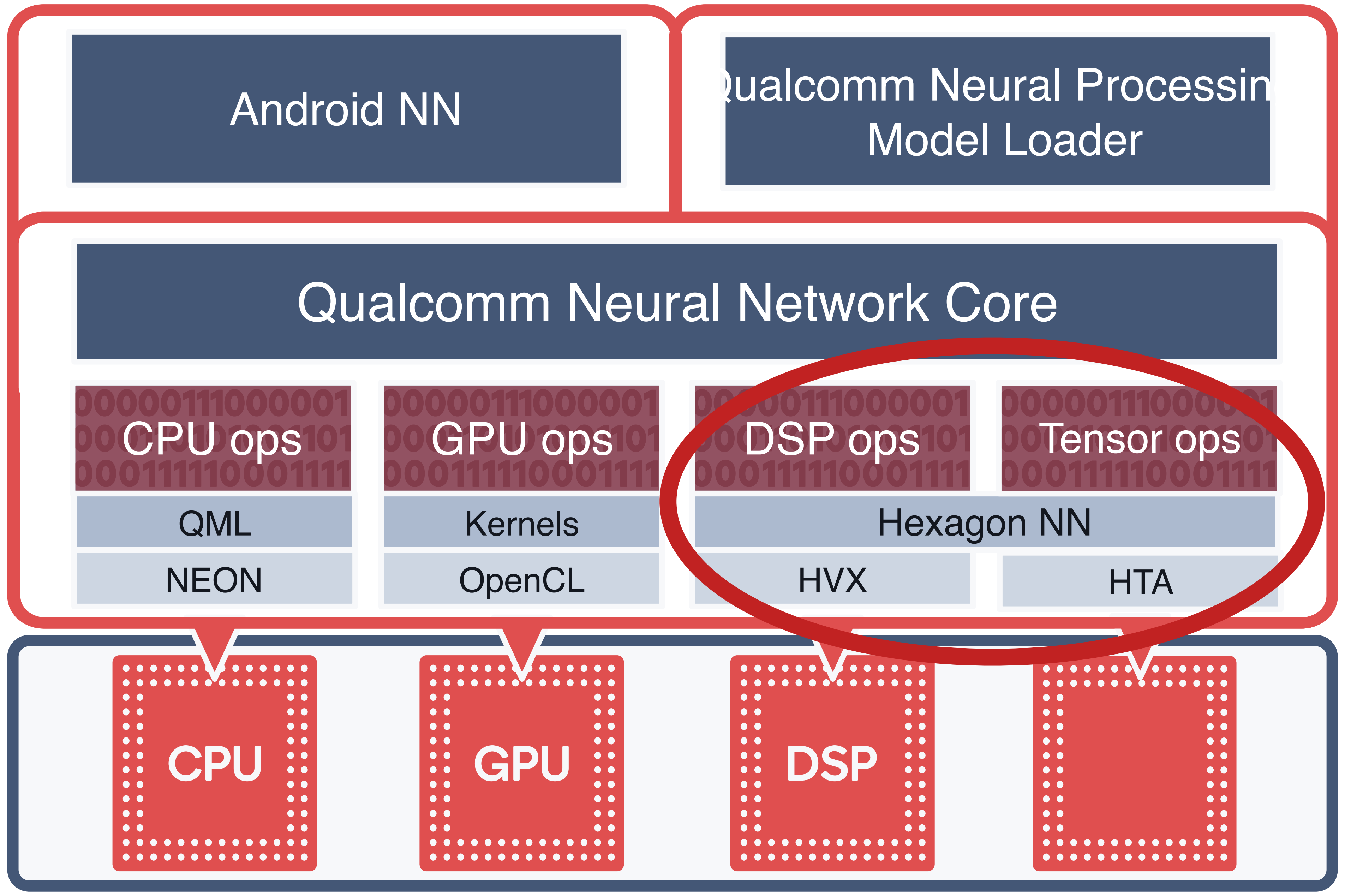
	Qualcomm Neural Processing SDK runtime	Android NNAPI library	Qualcomm® Hexagon ™ NN source & binary
Acceleration	🟥 HTA 🟥 DSP 🟥 GPU 🟥 CPU	🟥 HTA 🟥 DSP 🟥 GPU 🟦 CPU	🟥 HTA 🟥 DSP 🟦 GPU 🟦 CPU
Extendible, Partner I QTI	P □ Q ✓	P □ Q □	P ✓ Q ✓
Product input	TensorFlow ONNX Caffe2	Graph API, C++, or @TF-Lite	Graph API from DSP, C++ / HVX
Choose for	Fast experimentation Ease of migration Commercially proven Market leader	Accelerating other runtimes Future-proofing Cross-vendor	Low level access Flexibility & extensibility

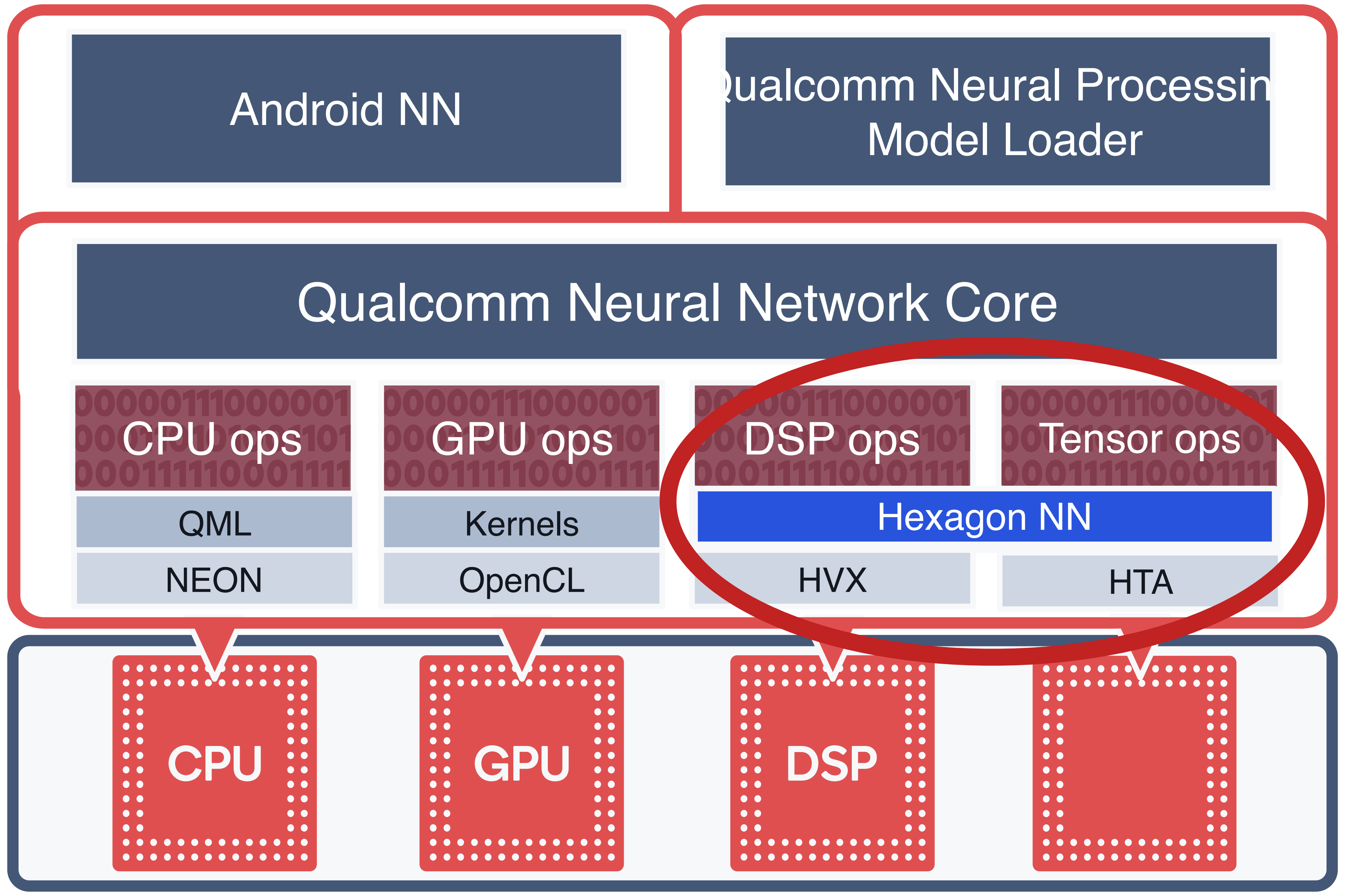


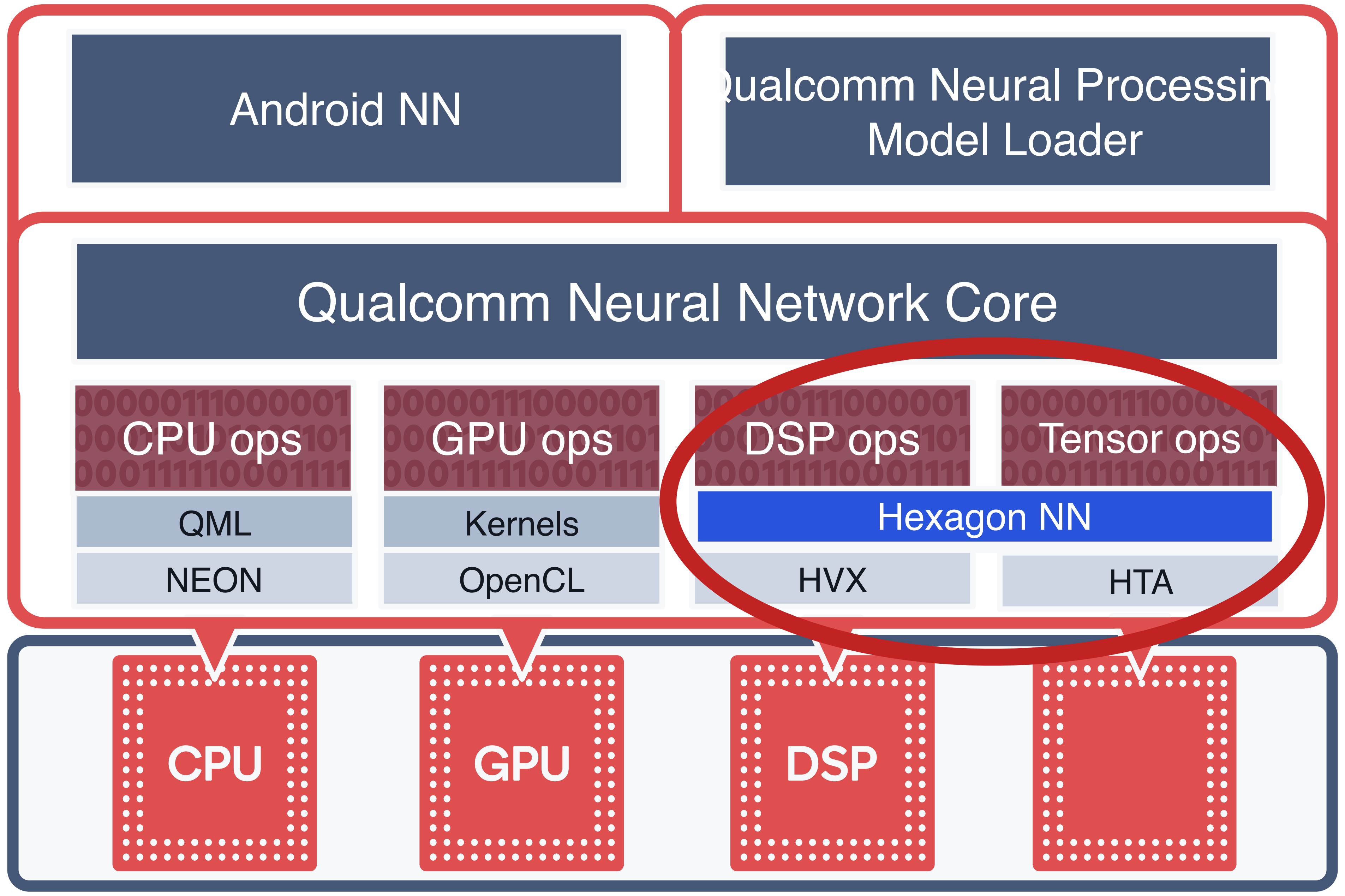
Our AI software products

Qualcomm Neural Processing SDK, Qualcomm Hexagon and Qualcomm AI Engine are products of Qualcomm Technologies, Inc.

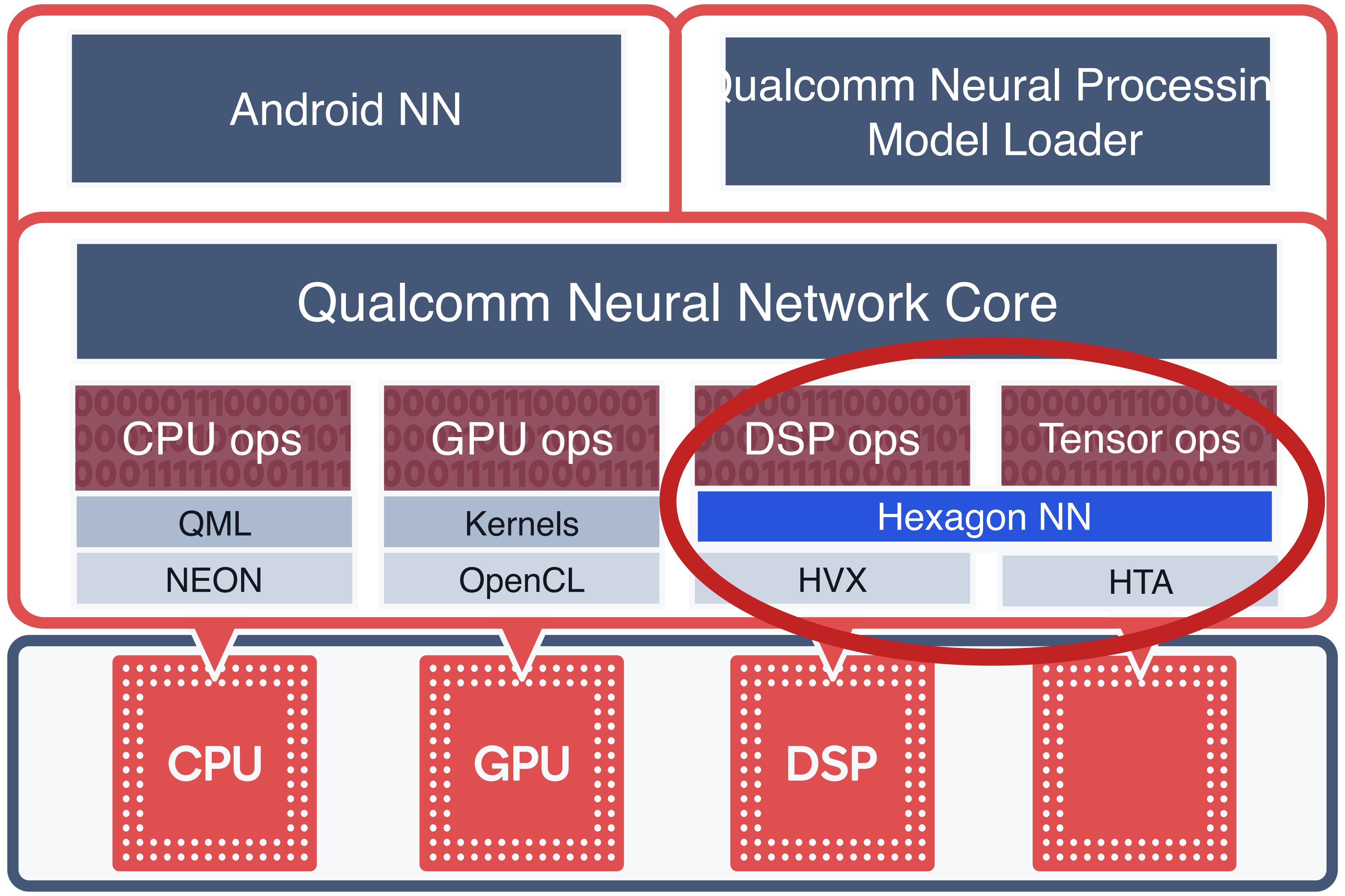

Android NN
ualcomm Neural Processin
Model Loader
Qualcomm Neural Network Core
CPU ops
GPU ops
DSP ops
Tensor ops
QML
Kernels
Hexagon NN
NEON
OpenCL
HVX
HTA
CPU
GPU
DSP


Android NN
ualcomm Neural Processin
Model Loader
Qualcomm Neural Network Core
CPU ops
GPU ops
DSP ops
Tensor ops
QML
Kernels
Hexagon NN
NEON
OpenCL
HVX
HTA
CPU
GPU
DSP


Android NN
ualcomm Neural Processin
Model Loader
Qualcomm Neural Network Core
CPU ops
GPU ops
DSP ops
Tensor ops
QML
Kernels
Hexagon NN
NEON
OpenCL
HVX
HTA
CPU
GPU
DSP

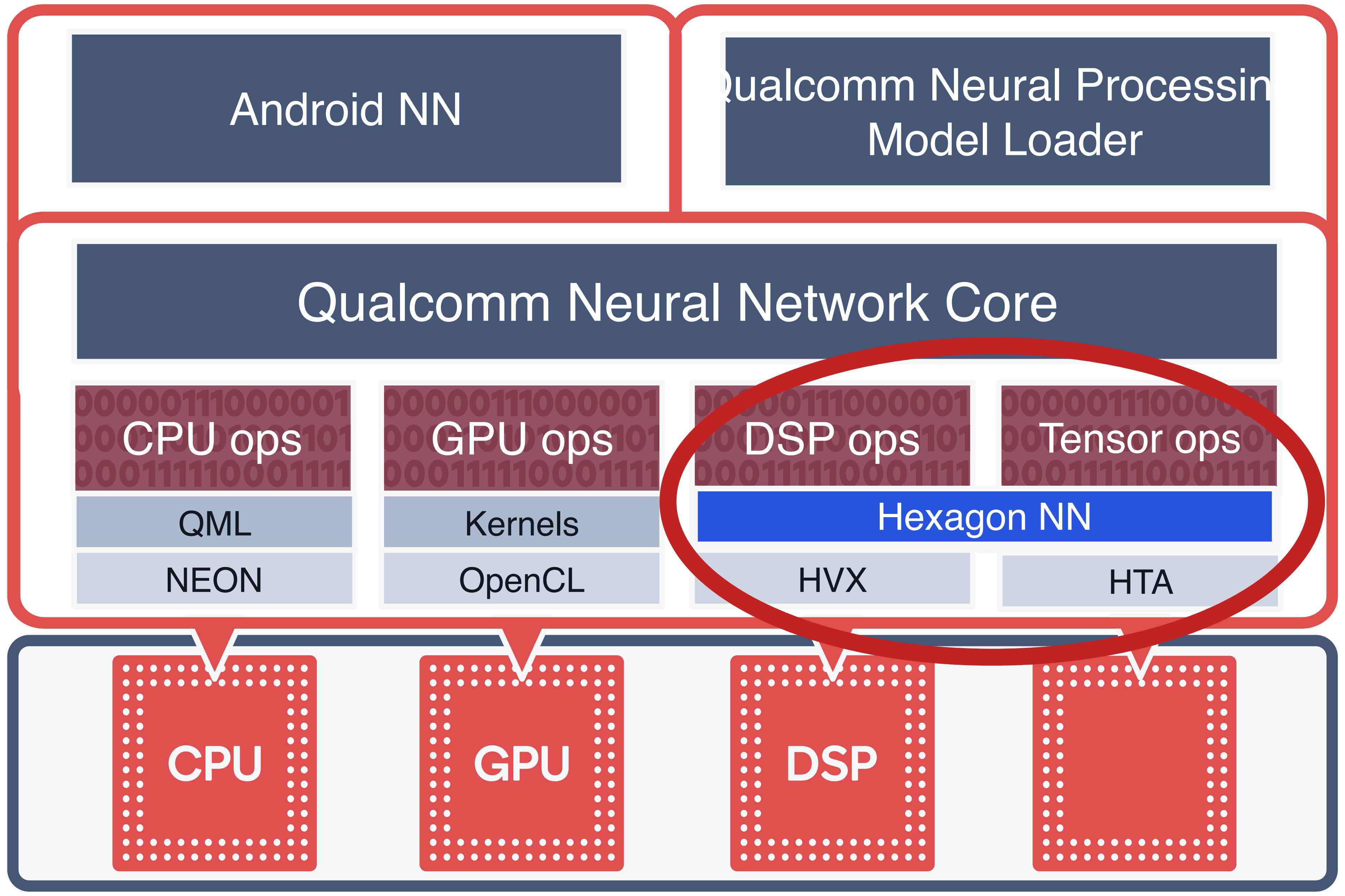

Android NN
ualcomm Neural Processin
Model Loader
Qualcomm Neural Network Core
CPU ops
GPU ops
DSP ops
Tensor ops
QML
Kernels
Hexagon NN
NEON
OpenCL
HVX
HTA
CPU
GPU
DSP

Hexagon NN



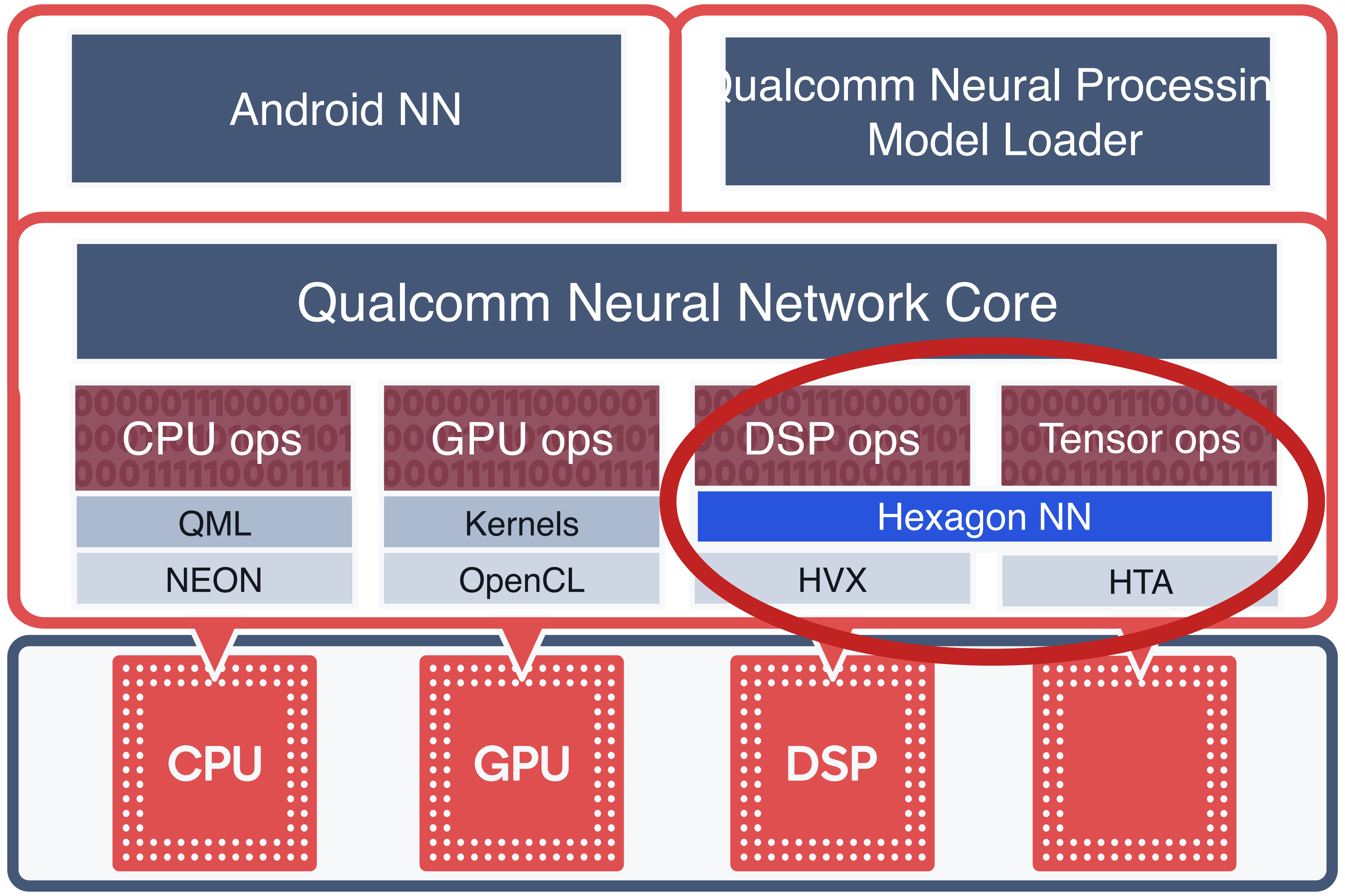
Hexagon NN

- Currently supports ~100 ops



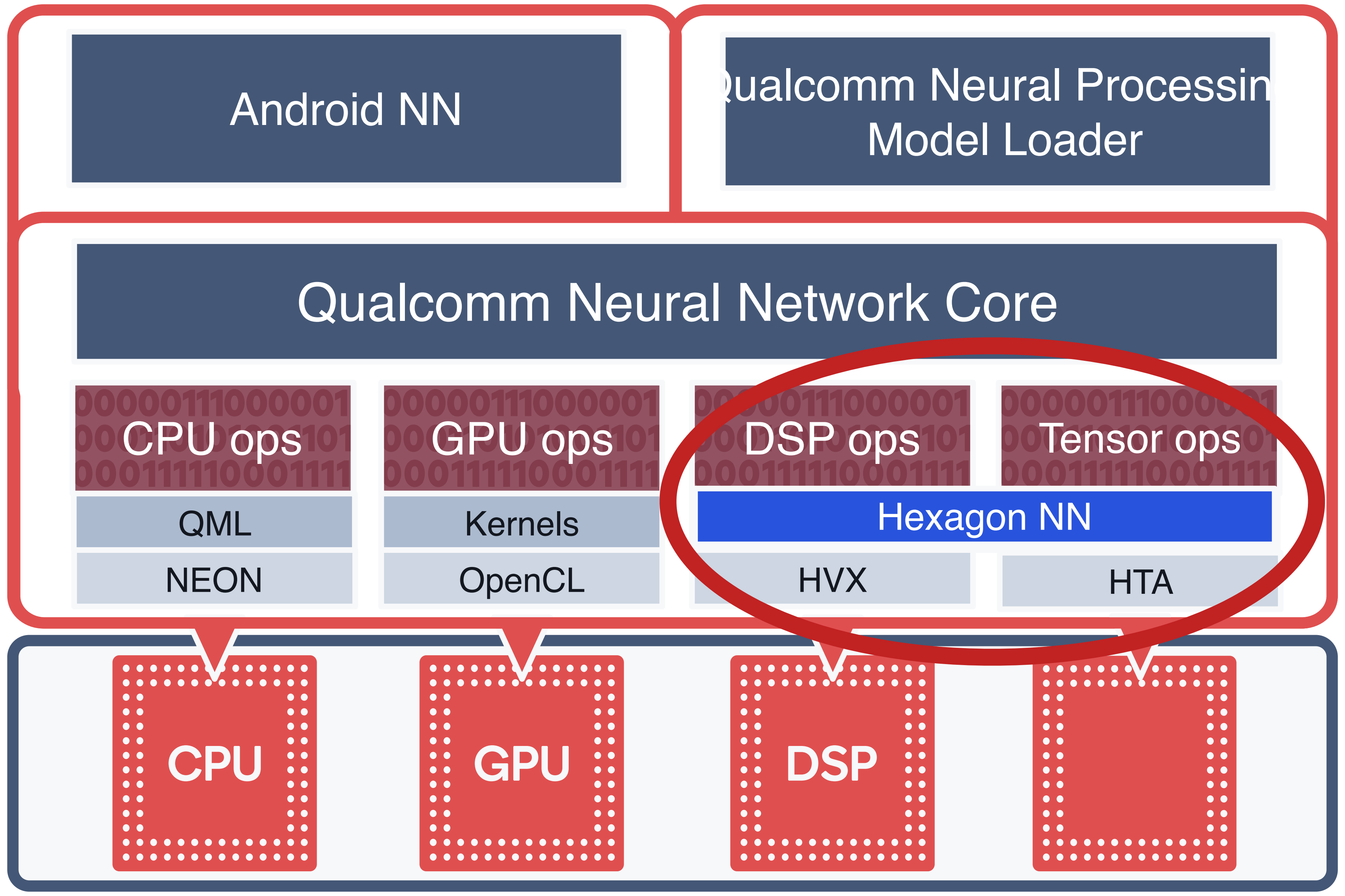
Hexagon NN

- Currently supports ~100 ops
- Handwritten and optimized across 3 different Hexagon architecture variations



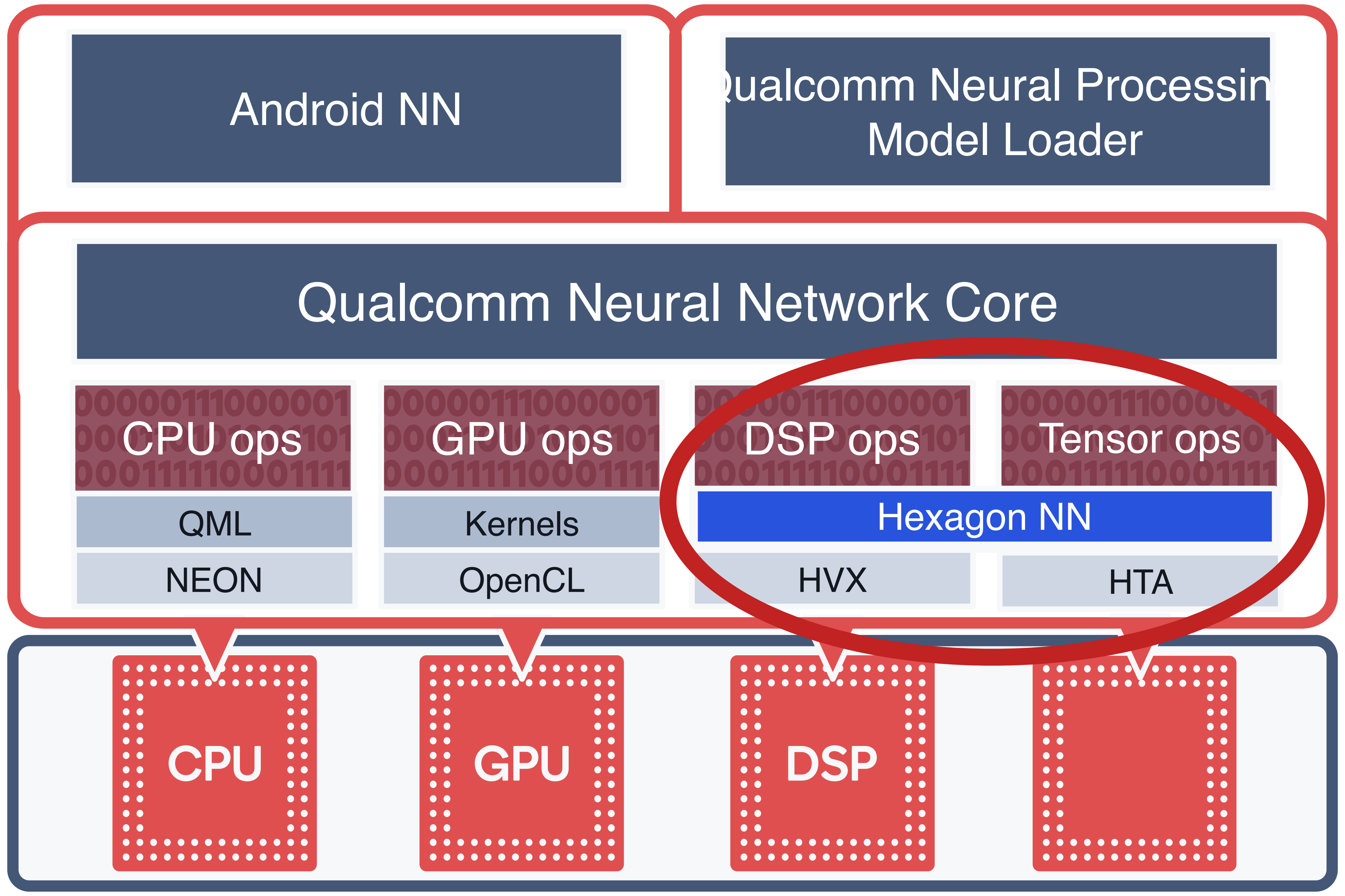
Hexagon NN

- Currently supports ~100 ops
- Handwritten and optimized across 3 different Hexagon architecture variations
- Ops have to be written for both Hexagon Vector Extensions (HVX) and Hexagon Tensor Accelerator (HTA) units



Hexagon NN

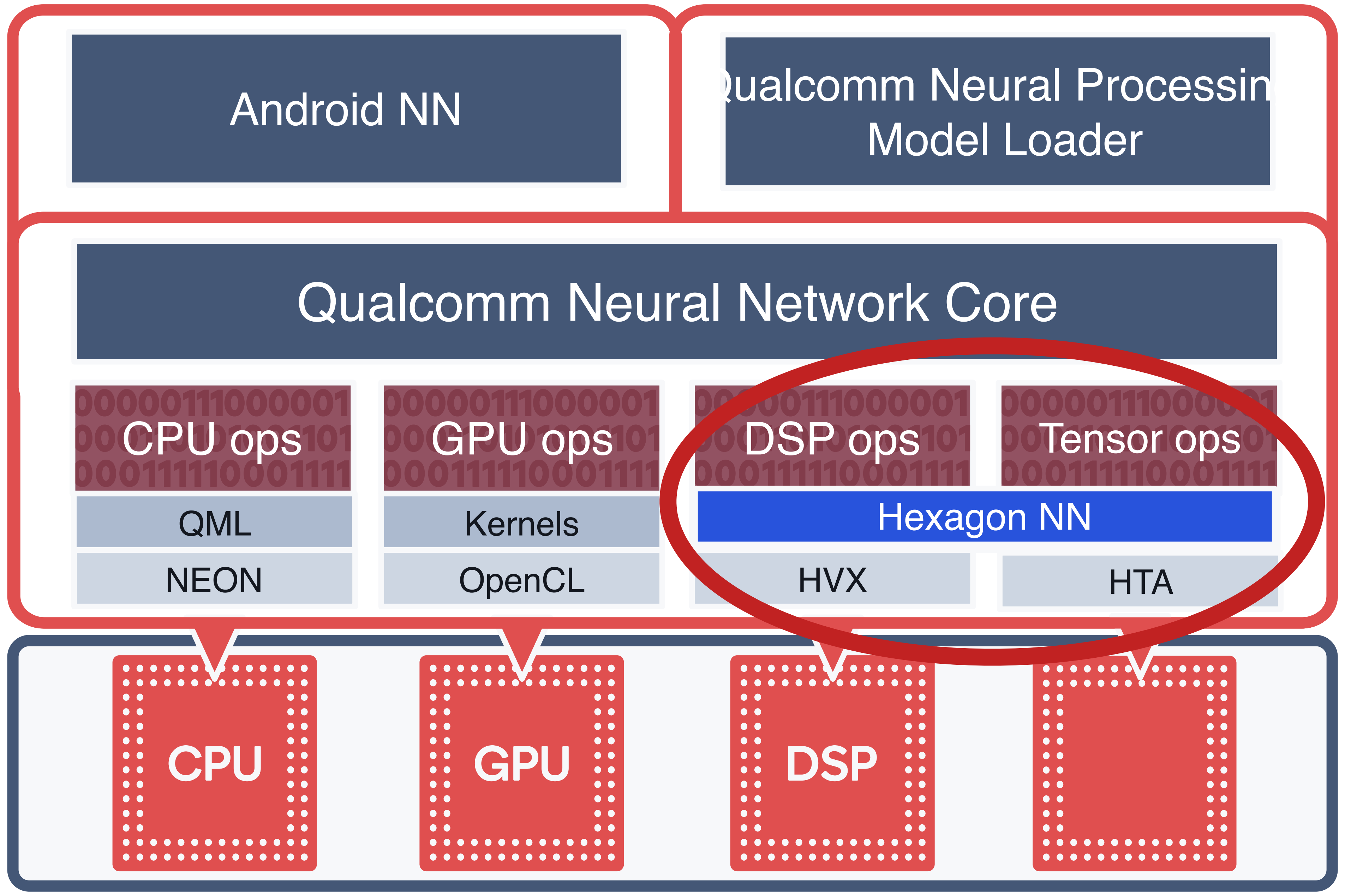
- Currently supports ~100 ops
- Handwritten and optimized across 3 different Hexagon architecture variations
- Ops have to be written for both Hexagon Vector Extensions (HVX) and Hexagon Tensor Accelerator (HTA) units
- Incredible demand from customers to add new operators and operator variants



Hexagon NN

- Currently supports ~100 ops
- Handwritten and optimized across 3 different Hexagon architecture variations
- Ops have to be written for both Hexagon Vector Extensions (HVX) and Hexagon Tensor Accelerator (HTA) units
- Incredible demand from customers to add new operators and operator variants

Hexagon is a flexible and power efficient but complex IP block to program efficiently. Like Halide for CV applications, TVM gives us internal development advantage and gives customers a tool to to develop custom operators.



Hexagon NN

- Currently supports ~100 ops
- Handwritten and optimized across 3 different Hexagon architecture variations
- Ops have to be written for both Hexagon Vector Extensions (HVX) and Hexagon Tensor Accelerator (HTA) units
- Incredible demand from customers to add new operators and operator variants

Hexagon is a flexible and power efficient but complex IP block to program efficiently. Like Halide for CV applications, TVM gives us internal development advantage and gives customers a tool to to develop custom operators.

TVM is key to ML Access on Hexagon

Key Ideas and Innovations

Qualcomm Technologies, Inc. is a leader in silicon for on-device and cloud solutions

Hexagon hardware provides a key power / performance advantage but is complicated to optimize

TVM and domain specific languages are key for per-kernel and whole graph optimization strategies

Our Qualcomm AI Research is advancing hardware aware optimization strategies

Qualcomm

Thank you

Follow us on: f 🐦 in ◎

For more information, visit us at:
www.qualcomm.com & www.qualcomm.com/blog

Nothing in these materials is an offer to sell any of the components or devices referenced herein.

References in this presentation to “Qualcomm” may mean Qualcomm Incorporated, Qualcomm Technologies, Inc., and/or other subsidiaries or business units within the Qualcomm corporate structure, as applicable. Qualcomm Incorporated includes Qualcomm’s licensing business, QTL, and the vast majority of its patent portfolio. Qualcomm Technologies, Inc., a wholly-owned subsidiary of Qualcomm Incorporated, operates, along with its subsidiaries, substantially all of Qualcomm’s engineering, research and development functions, and substantially all of its product and services businesses, including its semiconductor business, QCT.

Yida Wang

amazon

AWS AI

AWS AI

- The broadest and most complete set of machine learning capabilities
  - AI Services
  - Amazon SageMaker
  - ML Frameworks & Infrastructure

AWS AI

- The broadest and most complete set of machine learning capabilities
  - AI Services
  - Amazon SageMaker
  - ML Frameworks & Infrastructure
- More machine learning happens on AWS than anywhere else
  - 81% of deep learning in cloud runs on AWS

TVM@AWS

TVM@AWS

- As a cloud service: Amazon SageMaker Neo
  - Train models once, run anywhere with up to 2x performance improvement

TVM@AWS

- As a cloud service: Amazon SageMaker Neo
  - Train models once, run anywhere with up to 2x performance improvement
- As a solution
  - Fastest model inference on a number of Amazon EC2 instances
  - Alexa Wakeword model on Amazon Echo
  - Collaborating with a number of external device makers

TVM@AWS

- As a cloud service: Amazon SageMaker Neo
  - Train models once, run anywhere with up to 2x performance improvement
- As a solution
  - Fastest model inference on a number of Amazon EC2 instances
  - Alexa Wakeword model on Amazon Echo
  - Collaborating with a number of external device makers
- As a research project
  - Three accepted peer-reviewed papers
  - More under review and in preparation

TVM@AWS

- As a cloud service: Amazon SageMaker Neo
  - Train models once, run anywhere with up to 2x performance improvement
- As a solution
  - Fastest model inference on a number of Amazon EC2 instances
  - Alexa Wakeword model on Amazon Echo
  - Collaborating with a number of external device makers
- As a research project
  - Three accepted peer-reviewed papers
  - More under review and in preparation
- As a compiler
  - AWS Inferentia

AWS@TVM

AWS@TVM

- Join the effort from the very beginning, one of the major contributors

AWS@TVM

- Join the effort from the very beginning, one of the major contributors
- Major features in the past year
    - Frontend: TF object detection model
    - Relay: pass manager, VM, QNN dialect, graph partitioning
    - Optimization: vision-specific ops, conv2d_transpose, sparsity, BERT
    - Runtime: bring your own codegen

AWS@TVM

- Join the effort from the very beginning, one of the major contributors
- Major features in the past year
  - Frontend: TF object detection model
  - Relay: pass manager, VM, QNN dialect, graph partitioning
  - Optimization: vision-specific ops, conv2d_transpose, sparsity, BERT
  - Runtime: bring your own codegen
- Service in the community
  - 2 PMC members, 8 committers, 14 reviewers, and growing
  - Active participation and leadership

Jason Knight

OctoML

OctoML

Secure and efficient deep learning everywhere

Prediction:

Prediction:

N = number of people building machine learning models

Prediction:

N = number of people building machine learning models

M = number of software developers

Prediction:

N = number of people building machine learning models

M = number of software developers

N >> M

Prediction:

N = number of people building machine learning models

M = number of software developers

$$N >> M$$

as $t \rightarrow \infty$

Deep learning deployment should be easy.

For ***everyone***.

Deployment Pain/Complexity

- Model ingestion
- Performance estimation and comparison
- Cartesian product of models, frameworks, and hardware
- Optimization
  - O0, O1, O2
  - Target settings: march, mtune, mcpu
  - Size reductions
  - Quantization, pruning, distillation
- Custom operators (scheduling, cross hardware support)
- Lack of portability / varying coverage across frameworks
- Model integration
  - Output portability
  - Packaging (Android APK, iOS ipa, Python wheel, Maven artifact, etc)

Deep learning deployment should be easy.
For ***everyone***.

Deep learning deployment should be easy.
For ***everyone***.

TVM is core to making that happen.

Deep learning deployment should be easy.
For ***everyone***.

TVM is core to making that happen.

… but it’s only the first (important!) step

What are we doing about it?

To make DL deployment easy for everyone:

1. Strengthen the core:
   - Invest in open source TVM for robustness, accessibility, community, and coverage
   - (See next slide)

OctoML investments into TVM

OctoML invests in TVM

Talks today:

Unified IR – Tianqi Chen
Dynamic Execution and Virtual Machine – Jared Roesch and Haichen Shen
uTVM: TVM on bare-metal devices – Logan Weber
TVM at OctoML – Jason Knight

Not presented today:

TVM Transformer Improvements – Josh Fromm
Automatic Quantization – Ziheng Jiang

What are we doing about it?

To make DL deployment easy for everyone:

1. Strengthen the core:
    - ○ Invest in open source TVM for robustness, accessibility, community, and coverage
    - ○ (See next slide)

What are we doing about it?

To make DL deployment easy for everyone:

1. Strengthen the core:
   - Invest in open source TVM for robustness, accessibility, community, and coverage
   - (See next slide)

2. Build additional stepping stones
   - By forming a company! (come see our OctoML talk in the afternoon)

OctoML

Simple, secure, and efficient deployment of ML models in the edge and the cloud

Drive TVM adoption
Core infrastructure and improvements

Expand the set of users who can deploy ML models:
Services, automation, and integrations

Apache TVM ecosystem

OctoML

Team - The Octonauts

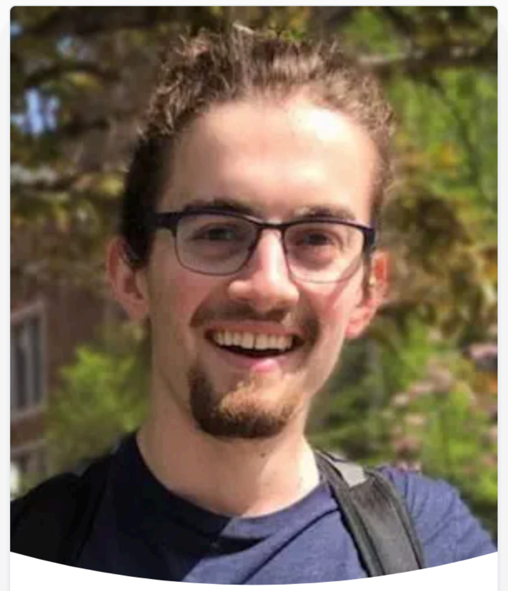
Luis Ceze
Co-founder, CEO
PhD in Computer Architecture and Compilers
Professor at UW-CSE
Venture Partner, Madrona Ventures

Jason Knight
Co-founder, CPO
PhD in Computational Biology and Machine Learning

Tianqi Chen
Co-founder, CTO
PhD in Machine Learning
Professor at CMU-CS

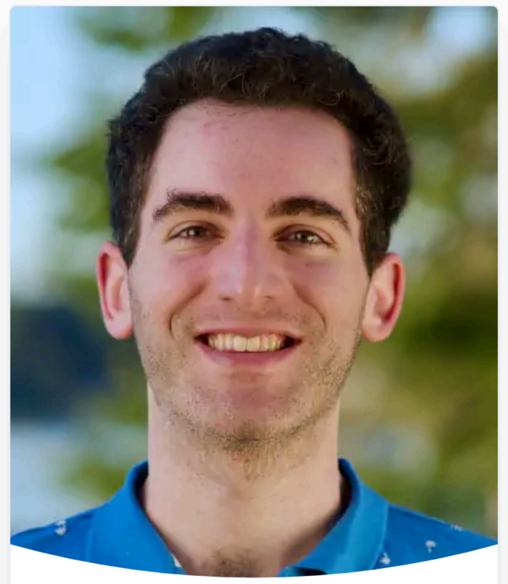
Thierry Moreau
Co-founder, Architect
PhD in Computer Architecture

Jared Roesch
Co-founder, Architect
(soon) PhD in Programming Languages



Logan Weber

An Wang



Josh Fromm

Zachary Tatlock

Andrew McHarg
Ziheng Jiang
Amanda Robles

Advisors

Jay Bartot

Carlos Guestrin

Arvind Krishnamurthy

tvm mxnet dmlc XGBoost nGraph

Find out more!

Come to our presentation about the Octomizer this afternoon

- Our first SaaS product for making DL deployment easy
  - Push button AutoTVM optimization
  - Perf comparisons/analysis across models, frameworks, and hardware
  - And more!

https://octoml.ai (mailing list signup)

@octoml on Twitter

Email us! (jknight@octoml.ai)

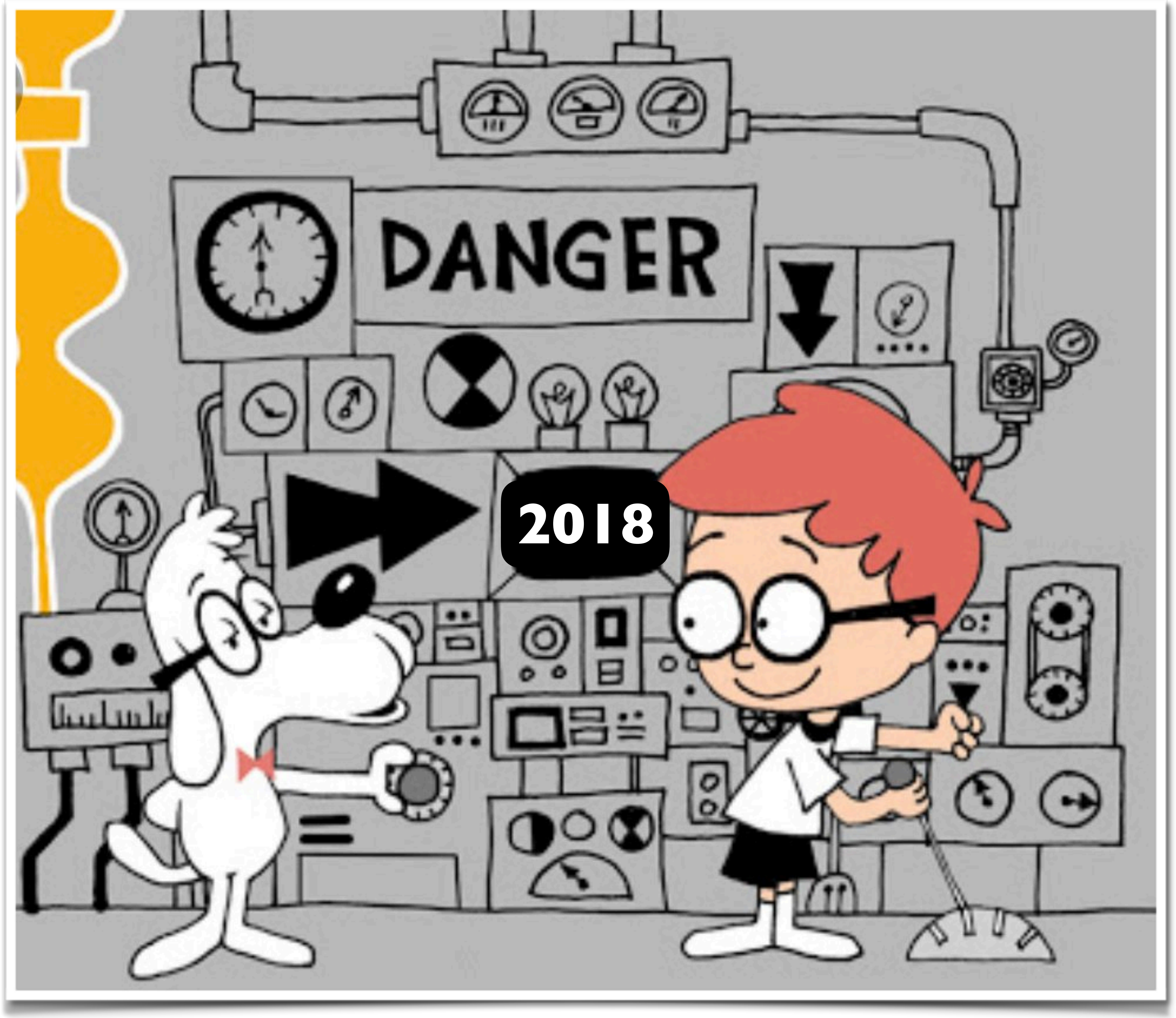
Zach Tatlock

PAUL G. ALLEN SCHOOL
OF COMPUTER SCIENCE & ENGINEERING

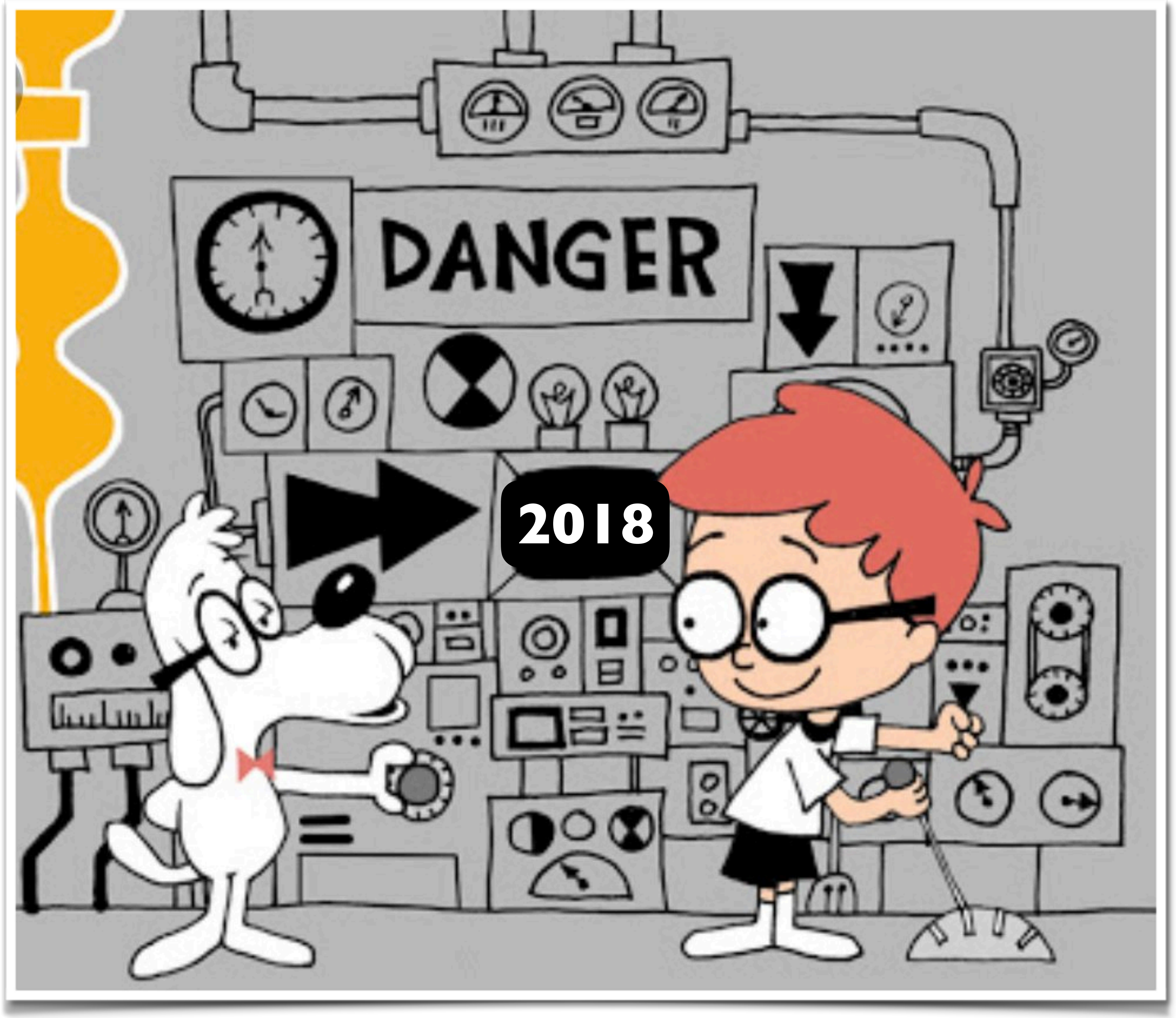
sampl

OctoML

Let's Get in the Wayback Machine



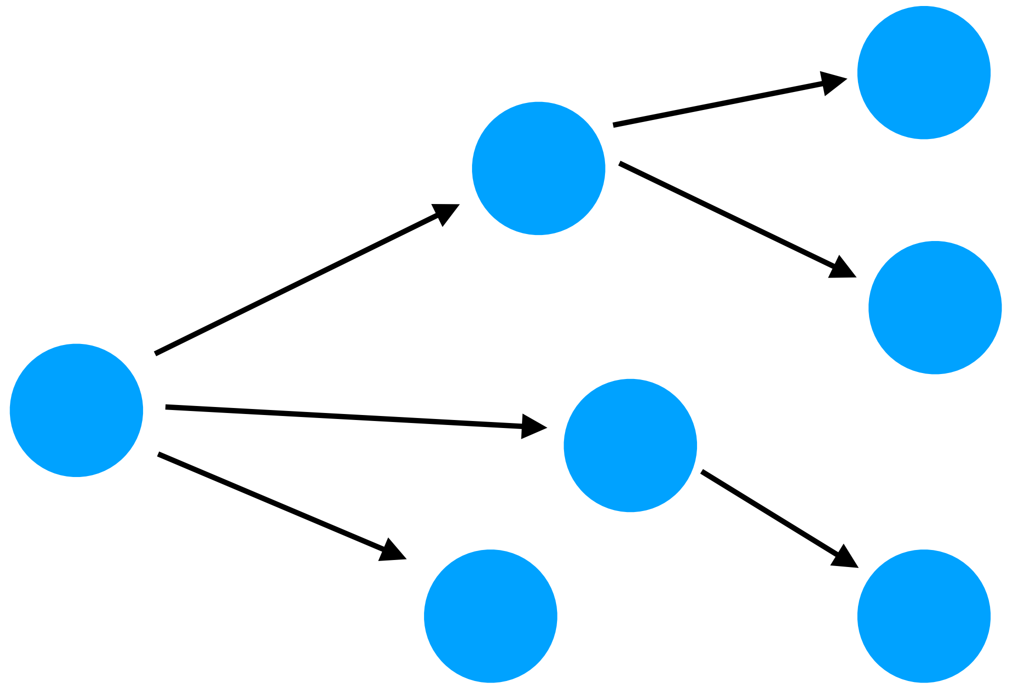
Let's Get in the Wayback Machine



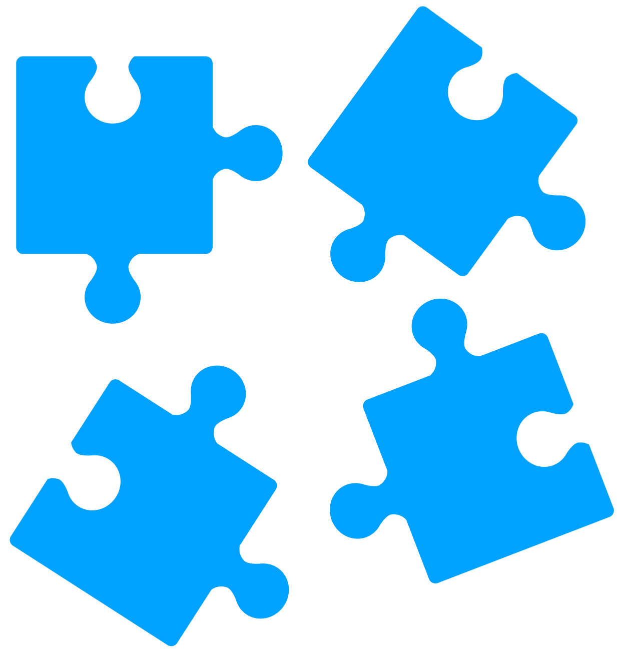
We apply the straightforward insight that machine learning models are just programs.
This generalization enables support for a greater range of programs, new optimizations, and the ability to targ… range of devices.
Excited about pro… …earch
http://sampl.cs.washington.edu
tvm
http://tvm.ai

Challenges for Deep Learning IRs

- State-of-the-art models increasingly depend on:
  - Datatypes - lists, trees, graphs
  - Control flow - branches, loops, recursion
  - Whole-program analyses and optimizations
- Any one feature "easy to bolt on"
- Folklore suggests full, expressive IR will be slow

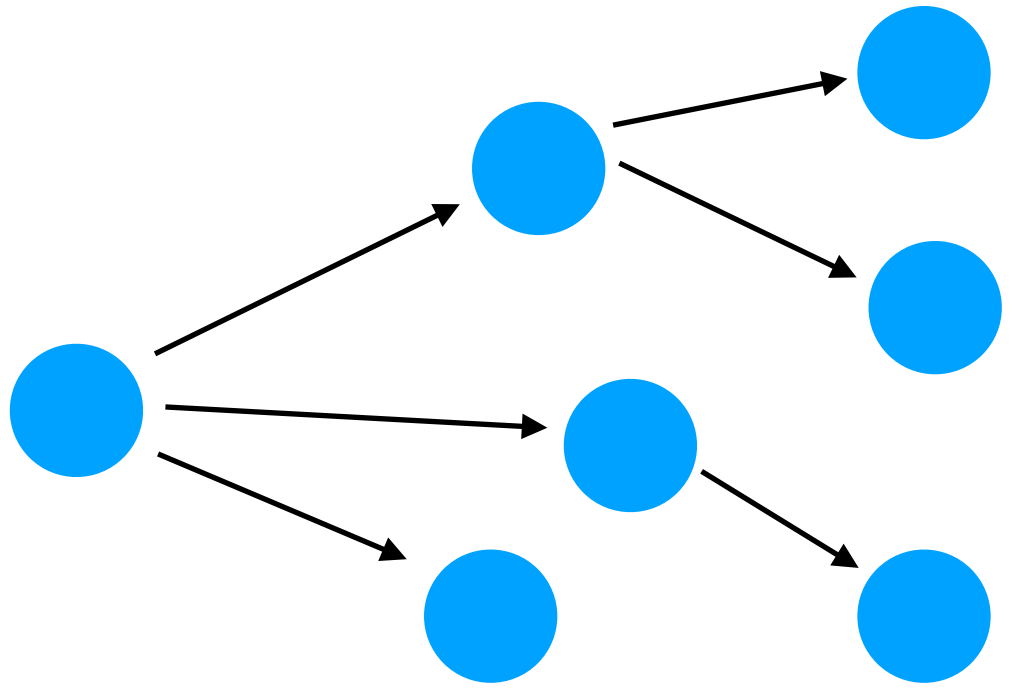


```
let encode = λ st.
    if(...):
        encode(step(st))
    else:
        ...
```

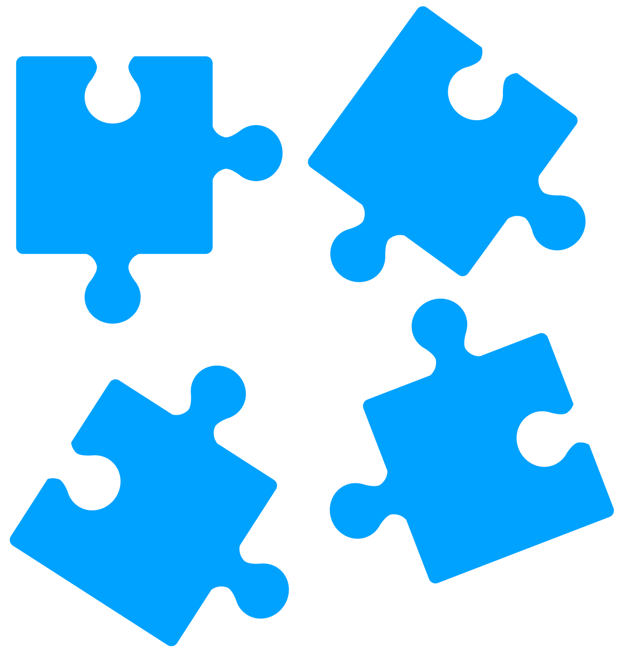


Challenges for Deep Learning IRs

- State-of-the-art models increasingly depend on:
  - Datatypes - lists, trees, graphs
  - Control flow - branches, loops, recursion
  - Whole-program analyses and optimizations
- Any one feature "easy to bolt on"
- Folklore suggests full, expressive IR will be slow

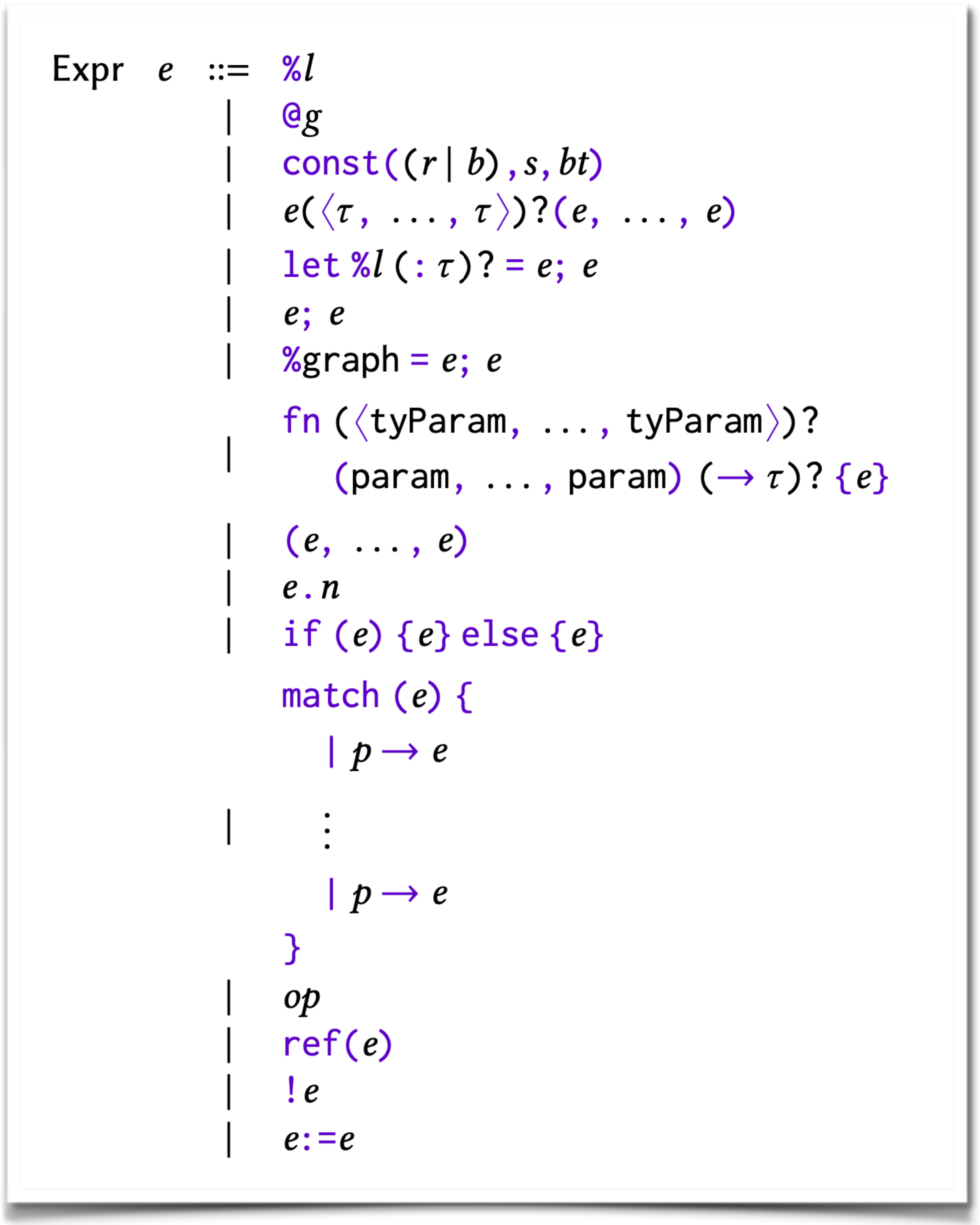


```
let encode = λ st.
    if(...):
        encode(step(st))
    else:
        ...
```



The Relay IR

- Relay generalizes NNVM
- Retains graph-level optimizations
- Provides more expressive features
    - Datatypes, control flow, code re-use
    - Functional semantics to simplify analysis
    - Automatic differentiation + optimizations

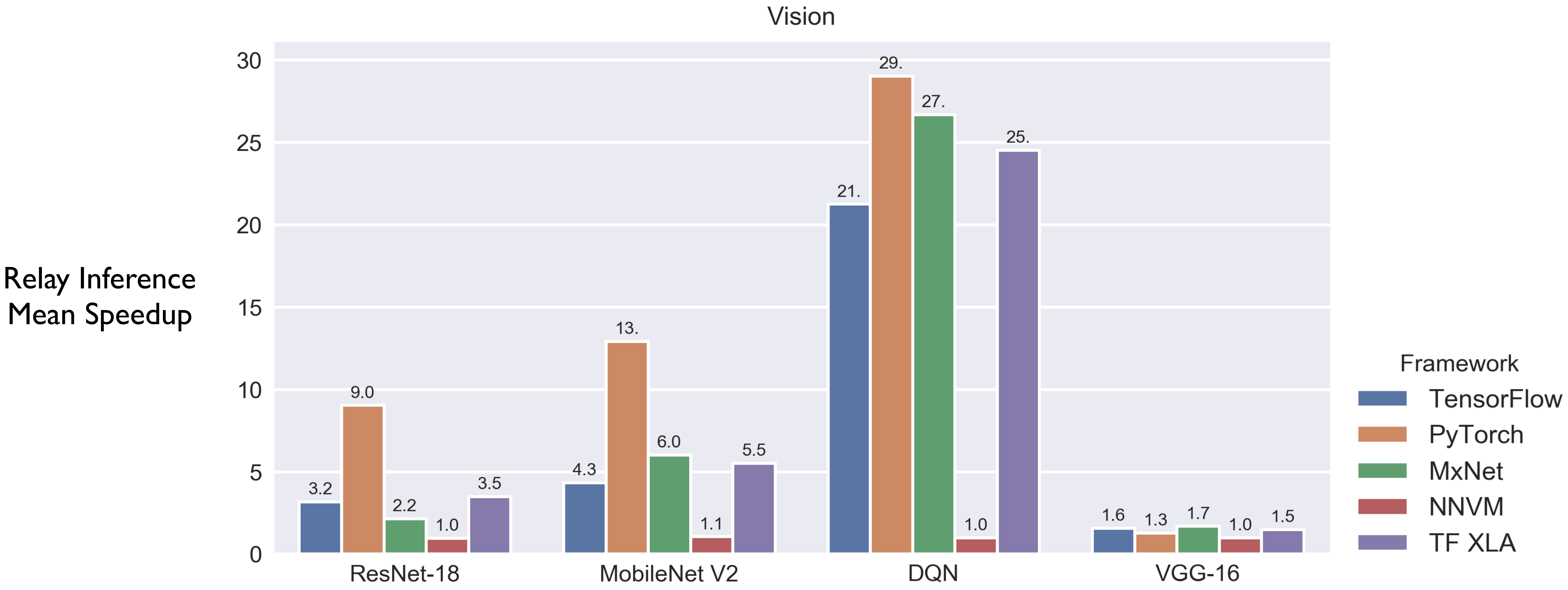


```
Expr  e  ::=  %l
           |  @g
           |  const((r | b), s, bt)
           |  e(⟨τ, ..., τ⟩)?(e, ..., e)
           |  let %l (: τ)? = e; e
           |  e; e
           |  %graph = e; e
              fn (⟨tyParam, ..., tyParam⟩)?
           |     (param, ..., param) (→ τ)? {e}
           |  (e, ..., e)
           |  e.n
           |  if (e) {e} else {e}
              match (e) {
                 | p → e
           |        ⋮
                 | p → e
              }
           |  op
           |  ref(e)
           |  !e
           |  e:=e
```

~ “OCaml for ML”

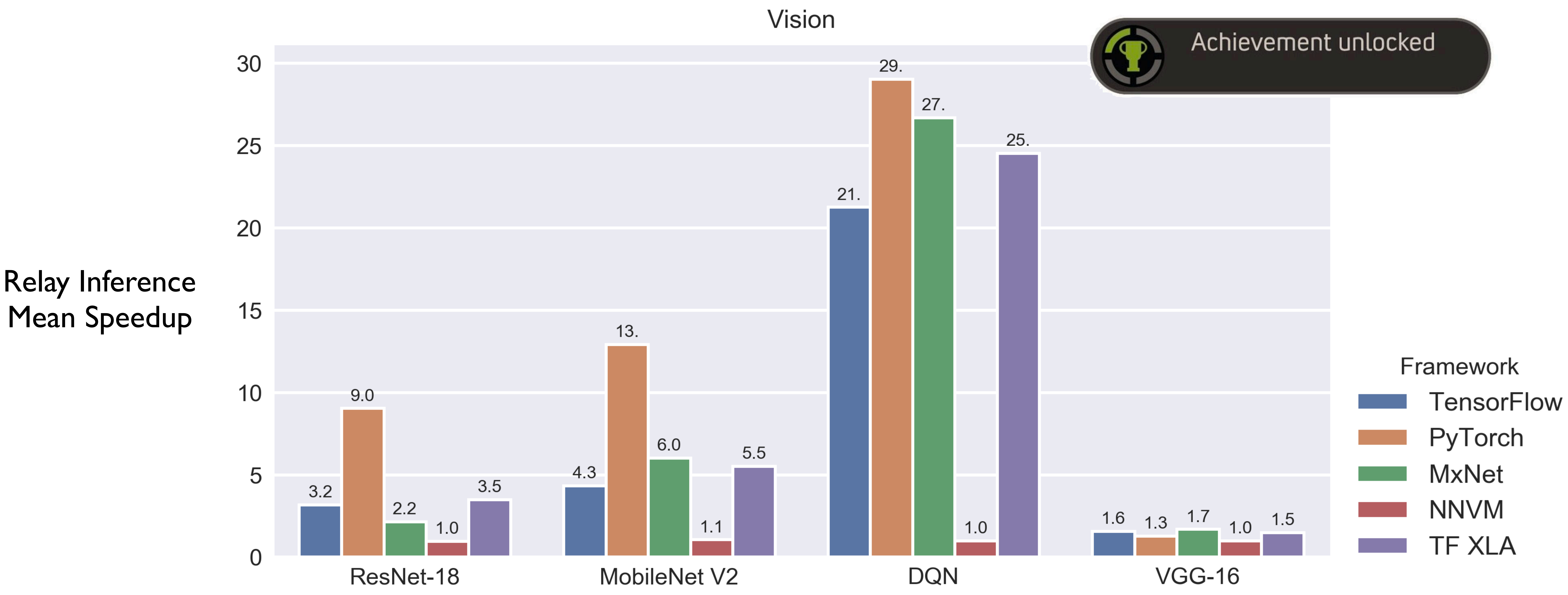
Relay: Expressiveness + Performance

- High-level Relay models match NNVM in traditional vision inference



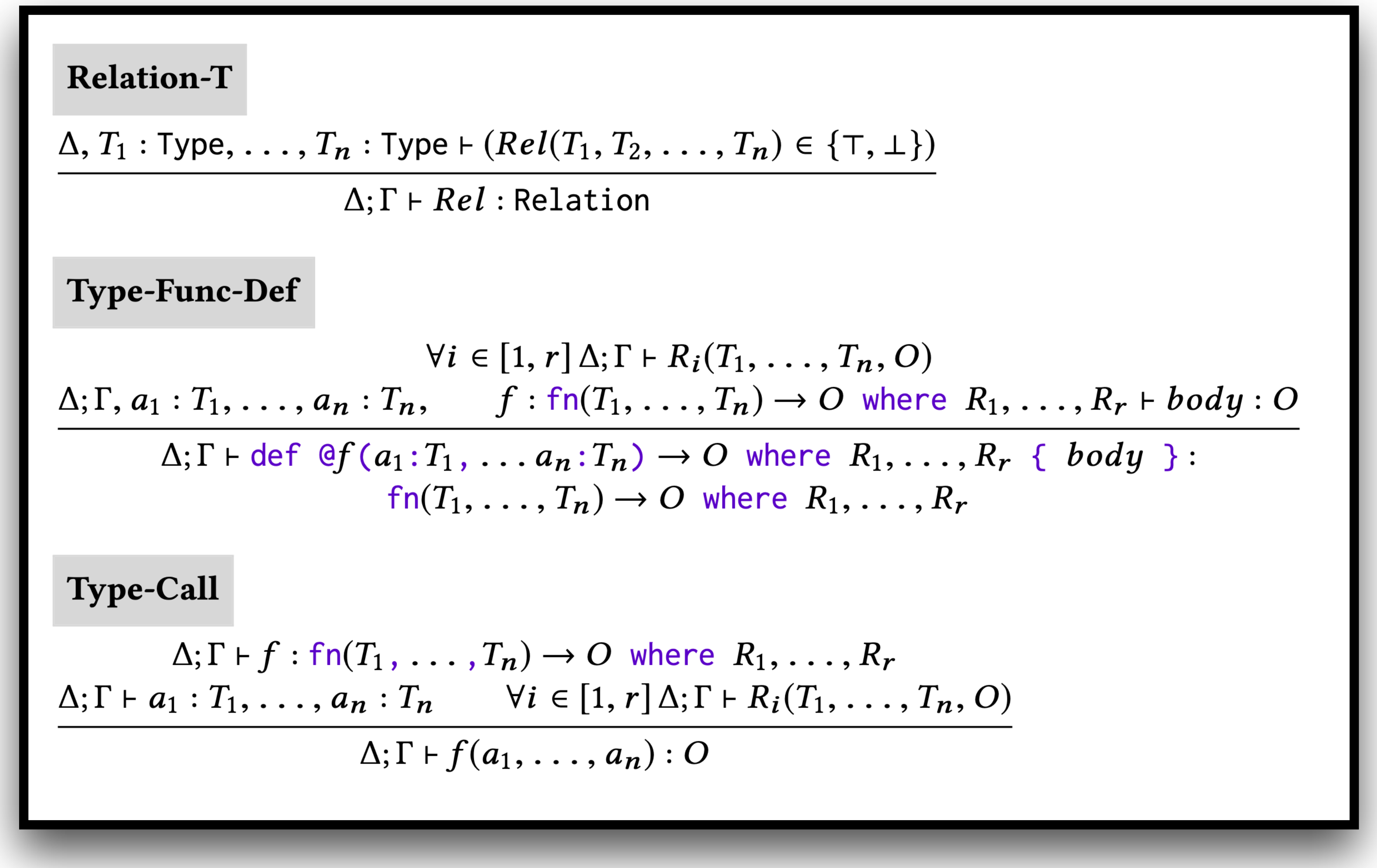
Relay: Expressiveness + Performance

- High-level Relay models match NNVM in traditional vision inference



Relay: Expressiveness + Performance

- Low-cost abstraction enabled by:
  - Tensor shape inference and specialization
  - High-level operator fusion
  - Whole-program partial evaluation



Relation-T

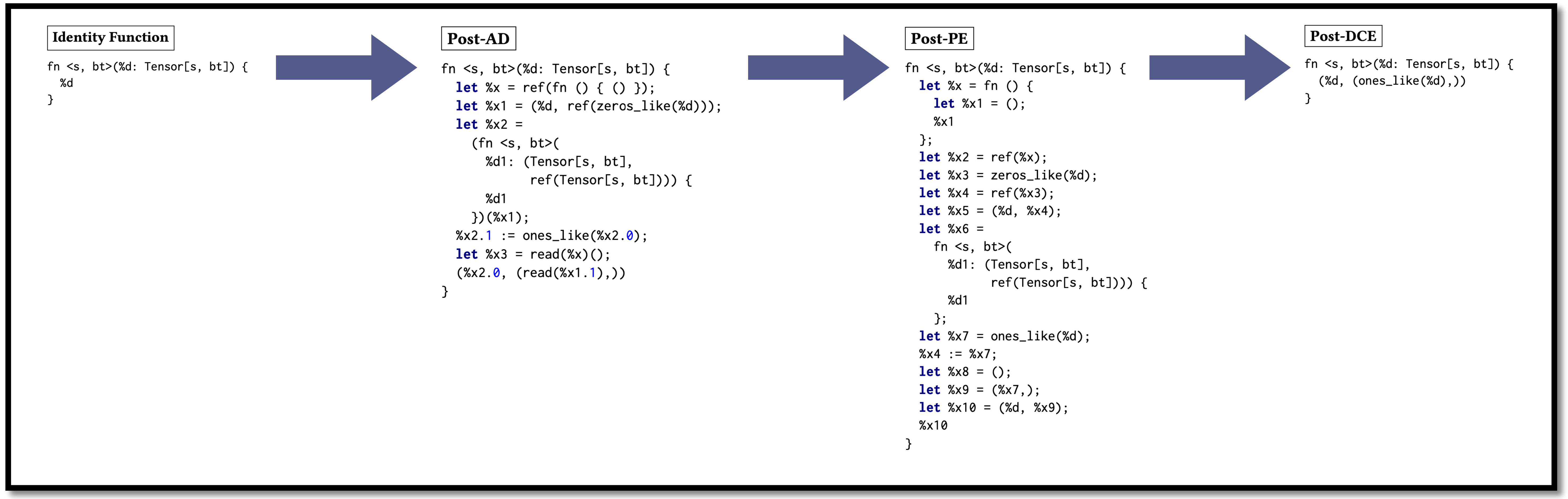
$$\frac{\Delta, T_1 : \mathsf{Type}, \ldots, T_n : \mathsf{Type} \vdash (Rel(T_1, T_2, \ldots, T_n) \in \{\top, \bot\})}{\Delta; \Gamma \vdash Rel : \mathtt{Relation}}$$

Type-Func-Def

$$\frac{\forall i \in [1, r]\, \Delta; \Gamma \vdash R_i(T_1, \ldots, T_n, O) \qquad \Delta; \Gamma, a_1 : T_1, \ldots, a_n : T_n, \quad f : \mathtt{fn}(T_1, \ldots, T_n) \to O \ \mathtt{where}\ R_1, \ldots, R_r \vdash body : O}{\Delta; \Gamma \vdash \mathtt{def}\ \mathtt{@}f\mathtt{(}a_1 \mathtt{:} T_1\mathtt{,} \ldots a_n \mathtt{:} T_n\mathtt{)} \to O \ \mathtt{where}\ R_1, \ldots, R_r \ \mathtt{\{}\ body\ \mathtt{\}} : \mathtt{fn}(T_1, \ldots, T_n) \to O \ \mathtt{where}\ R_1, \ldots, R_r}$$

Type-Call

$$\frac{\Delta; \Gamma \vdash f : \mathtt{fn}(T_1\mathtt{,} \ldots\mathtt{,} T_n) \to O \ \mathtt{where}\ R_1, \ldots, R_r \qquad \Delta; \Gamma \vdash a_1 : T_1, \ldots, a_n : T_n \qquad \forall i \in [1, r]\, \Delta; \Gamma \vdash R_i(T_1, \ldots, T_n, O)}{\Delta; \Gamma \vdash f(a_1, \ldots, a_n) : O}$$



Identity Function

```
fn <s, bt>(%d: Tensor[s, bt]) {
  %d
}
```

Post-AD

```
fn <s, bt>(%d: Tensor[s, bt]) {
  let %x = ref(fn () { () });
  let %x1 = (%d, ref(zeros_like(%d)));
  let %x2 =
    (fn <s, bt>(
      %d1: (Tensor[s, bt],
            ref(Tensor[s, bt]))) {
      %d1
    })(%x1);
  %x2.1 := ones_like(%x2.0);
  let %x3 = read(%x)();
  (%x2.0, (read(%x1.1),))
}
```

Post-PE

```
fn <s, bt>(%d: Tensor[s, bt]) {
  let %x = fn () {
    let %x1 = ();
    %x1
  };
  let %x2 = ref(%x);
  let %x3 = zeros_like(%d);
  let %x4 = ref(%x3);
  let %x5 = (%d, %x4);
  let %x6 =
    fn <s, bt>(
      %d1: (Tensor[s, bt],
            ref(Tensor[s, bt]))) {
      %d1
    };
  let %x7 = ones_like(%d);
  %x4 := %x7;
  let %x8 = ();
  let %x9 = (%x7,);
  let %x10 = (%d, %x9);
  %x10
}
```

Post-DCE

```
fn <s, bt>(%d: Tensor[s, bt]) {
  (%d, (ones_like(%d),))
}
```

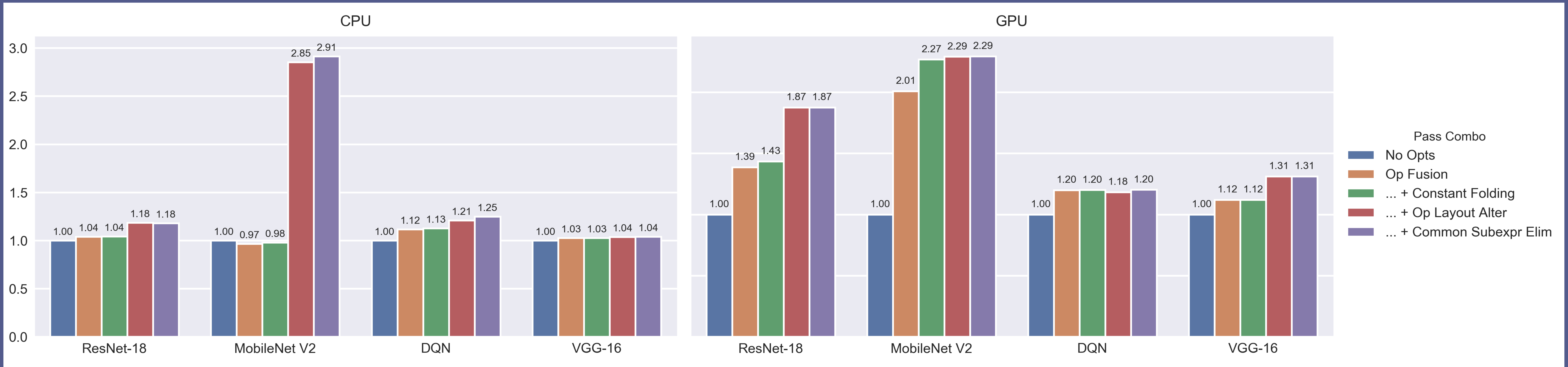
Relay: Expressiveness + Performance

- Low-cost abstraction enabled by:

Relation-T

$\Delta, T_1 : \mathsf{Type}, \ldots, T_n : \mathsf{Type} \vdash (Rel(T_1, T_2, \ldots, T_n) \in \{\top, \bot\})$

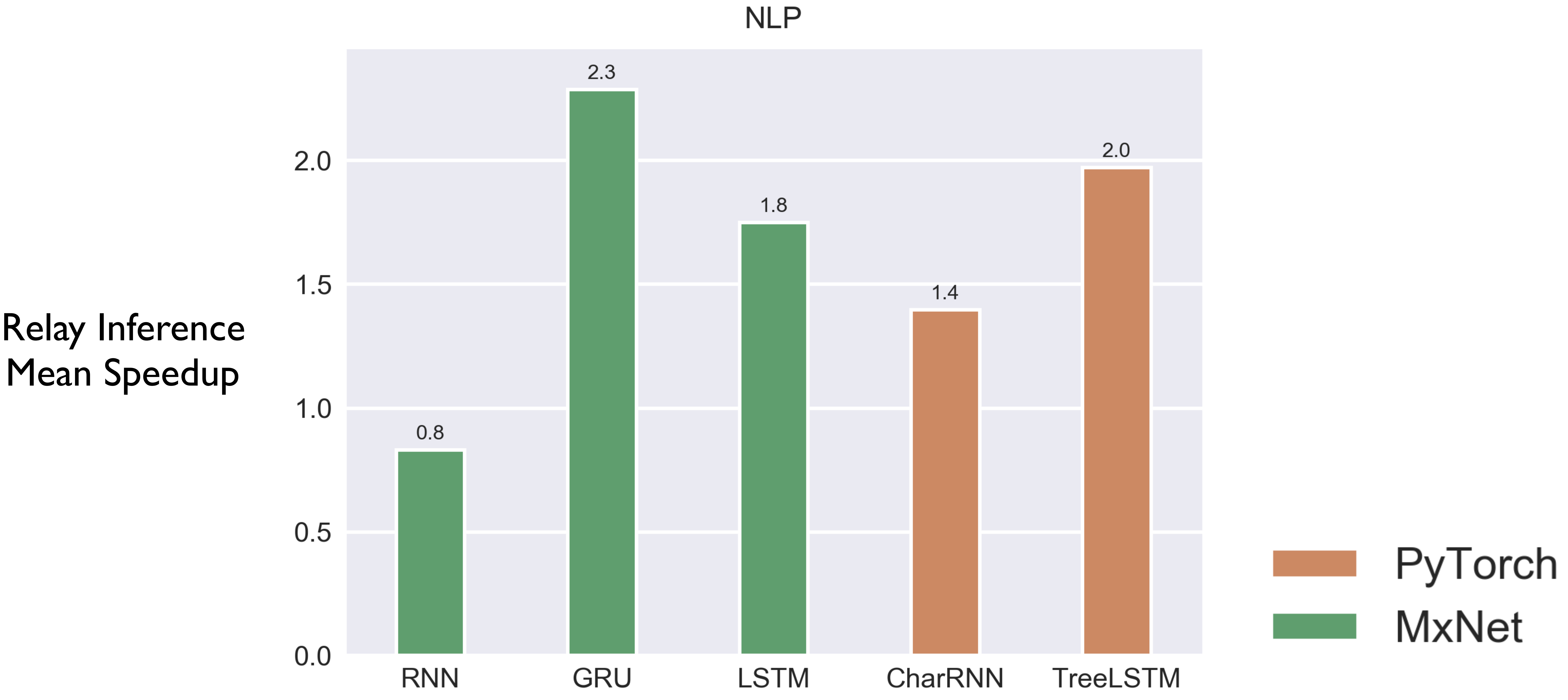
But most of all by extensible, composable optimization framework!



```
    %x10
}
```

Relay Win: Support for New Models

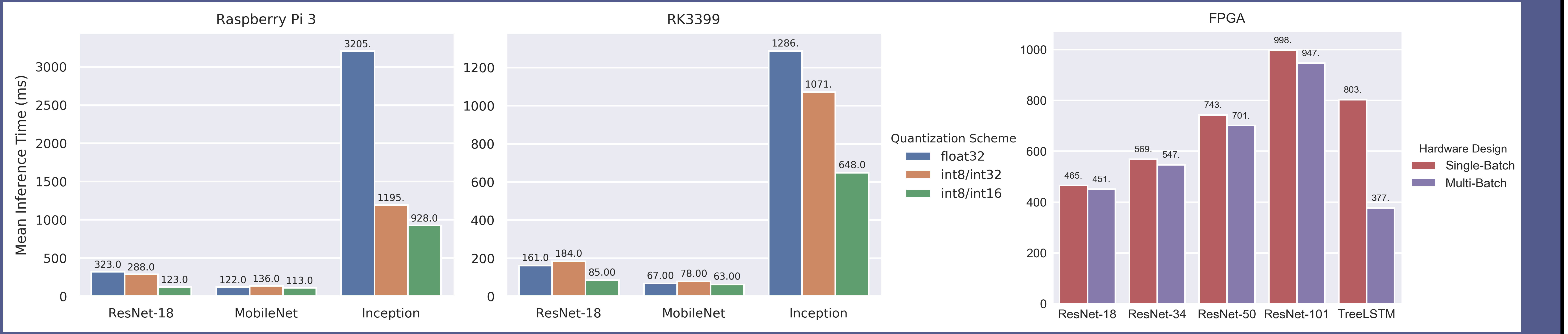
- High-level Relay models for RNNs and LSTMs can outperform the rest



Relay Win: Support for New Models

- High-level Relay models for RNNs and LSTMs can outperform the rest

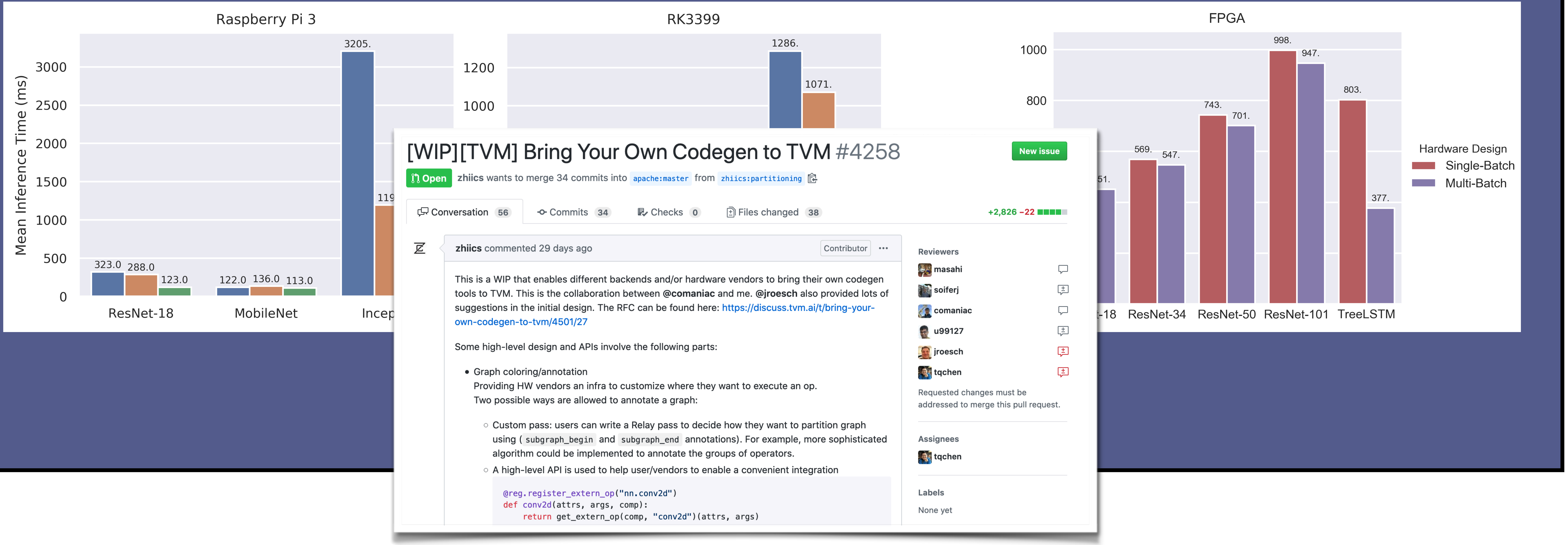
Plus support for new/improved targets via high-level transformations:



Relay Win: Support for New Models

- High-level Relay models for RNNs and LSTMs can outperform the rest

Plus support for new/improved targets via high-level transformations:



[WIP][TVM] Bring Your Own Codegen to TVM #4258

zhiics wants to merge 34 commits into `apache:master` from `zhiics:partitioning`

zhiics commented 29 days ago

This is a WIP that enables different backends and/or hardware vendors to bring their own codegen tools to TVM. This is the collaboration between **@comaniac** and me. **@jroesch** also provided lots of suggestions in the initial design. The RFC can be found here: https://discuss.tvm.ai/t/bring-your-own-codegen-to-tvm/4501/27

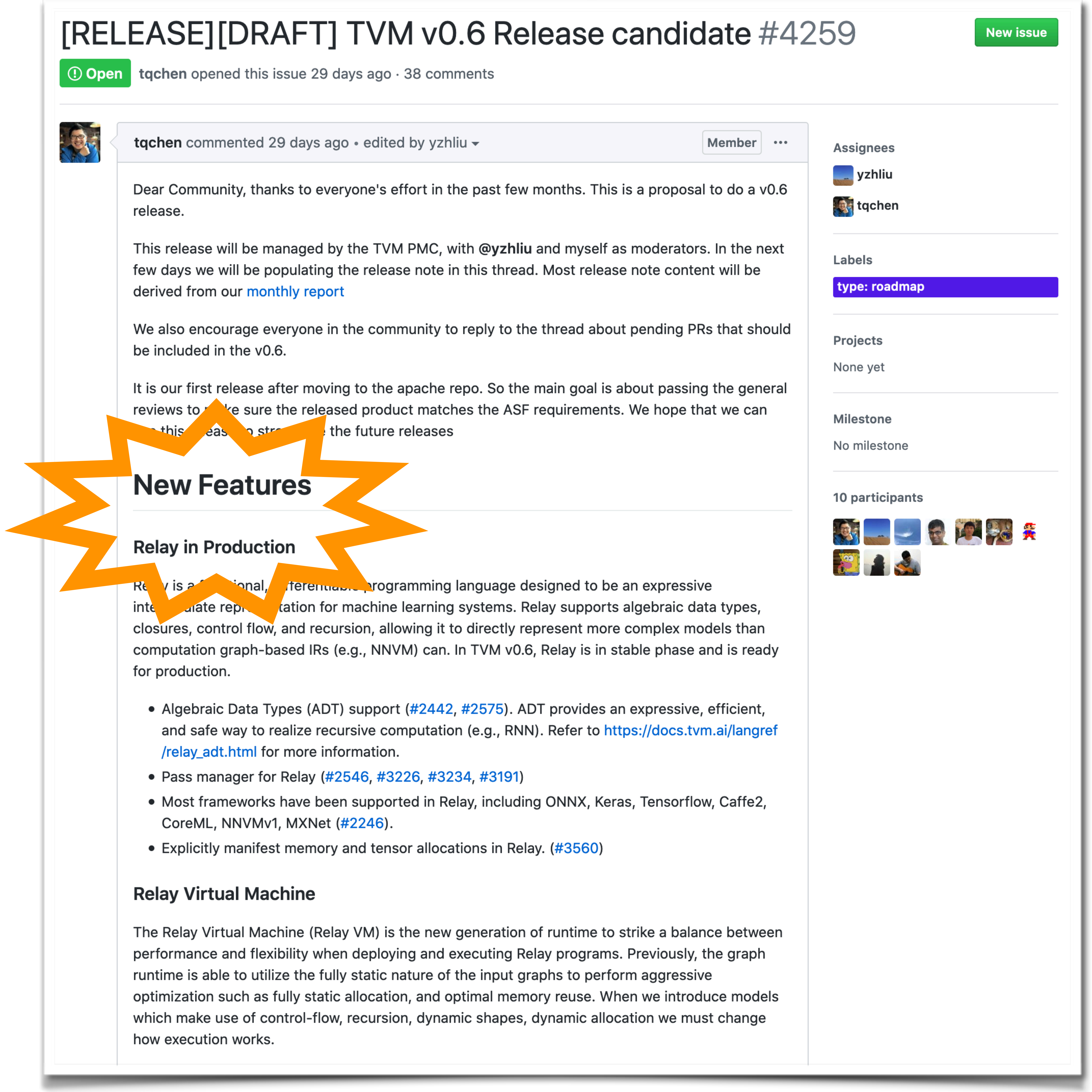
Some high-level design and APIs involve the following parts:

- Graph coloring/annotation
  Providing HW vendors an infra to customize where they want to execute an op.
  Two possible ways are allowed to annotate a graph:
  - Custom pass: users can write a Relay pass to decide how they want to partition graph using (`subgraph_begin` and `subgraph_end` annotations). For example, more sophisticated algorithm could be implemented to annotate the groups of operators.
  - A high-level API is used to help user/vendors to enable a convenient integration

```
@reg.register_extern_op("nn.conv2d")
def conv2d(attrs, args, comp):
    return get_extern_op(comp, "conv2d")(attrs, args)
```

Requested changes must be addressed to merge this pull request.

Research Ready ➡ Production Ready



[RELEASE][DRAFT] TVM v0.6 Release candidate #4259

tqchen opened this issue 29 days ago · 38 comments

tqchen commented 29 days ago • edited by yzhliu

Dear Community, thanks to everyone's effort in the past few months. This is a proposal to do a v0.6 release.

This release will be managed by the TVM PMC, with **@yzhliu** and myself as moderators. In the next few days we will be populating the release note in this thread. Most release note content will be derived from our monthly report

We also encourage everyone in the community to reply to the thread about pending PRs that should be included in the v0.6.

It is our first release after moving to the apache repo. So the main goal is about passing the general reviews to ke sure the released product matches the ASF requirements. We hope that we can this eas o str the future releases

New Features

Relay in Production

Re y is a ional, ferentia programming language designed to be an expressive inte iate repr tation for machine learning systems. Relay supports algebraic data types, closures, control flow, and recursion, allowing it to directly represent more complex models than computation graph-based IRs (e.g., NNVM) can. In TVM v0.6, Relay is in stable phase and is ready for production.

- Algebraic Data Types (ADT) support (#2442, #2575). ADT provides an expressive, efficient, and safe way to realize recursive computation (e.g., RNN). Refer to https://docs.tvm.ai/langref/relay_adt.html for more information.
- Pass manager for Relay (#2546, #3226, #3234, #3191)
- Most frameworks have been supported in Relay, including ONNX, Keras, Tensorflow, Caffe2, CoreML, NNVMv1, MXNet (#2246).
- Explicitly manifest memory and tensor allocations in Relay. (#3560)

Relay Virtual Machine

The Relay Virtual Machine (Relay VM) is the new generation of runtime to strike a balance between performance and flexibility when deploying and executing Relay programs. Previously, the graph runtime is able to utilize the fully static nature of the input graphs to perform aggressive optimization such as fully static allocation, and optimal memory reuse. When we introduce models which make use of control-flow, recursion, dynamic shapes, dynamic allocation we must change how execution works.

Relay + You!

- Relay merged in to TVM mainline
  - Documentation, tutorials, examples
  - Add your own analyses and optimizations
  - Target new accelerators
  - Support new models
  - Tons of community support!

+ many more amazing folks!

Relay + You!

- Relay merged in to TVM mainline
  - Documentation, tutorials, examples
  - Add your own analyses and optimizations
  - Target new accelerators
  - Support new models
  - Tons of community support!

+ many more amazing folks!

Tianqi Chen

PAUL G. ALLEN SCHOOL
OF COMPUTER SCIENCE & ENGINEERING

sampl

OctoML

Current Deep Learning Landscape

Frameworks and Inference engines

ONNX RUNTIME

TensorRT

DL Compilers

nGraph

GLOW

MLIR

Kenrel Libraries

CuDNN NNPack MKL-DNN

Hand optimized

Hardware

Current Deep Learning Landscape

Frameworks and Inference engines

ONNX RUNTIME

TensorRT

DL Compilers

nGraph

GLOW

MLIR

Kenrel Libraries

CuDNN NNPack MKL-DNN

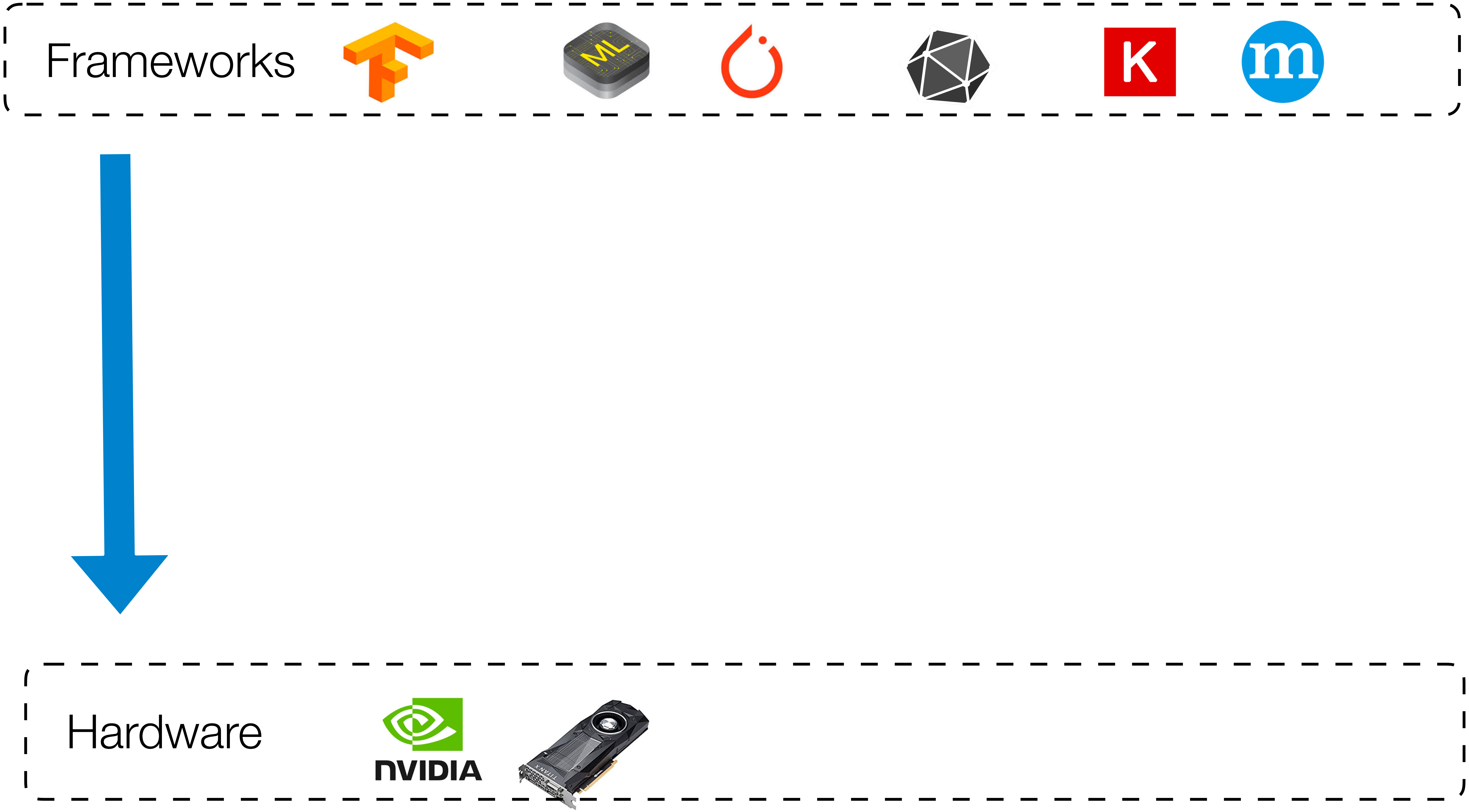
Hand optimized

Hardware

tvm

Open source, automated end-to-end optimization framework for deep learning.

Existing Deep Learning Frameworks

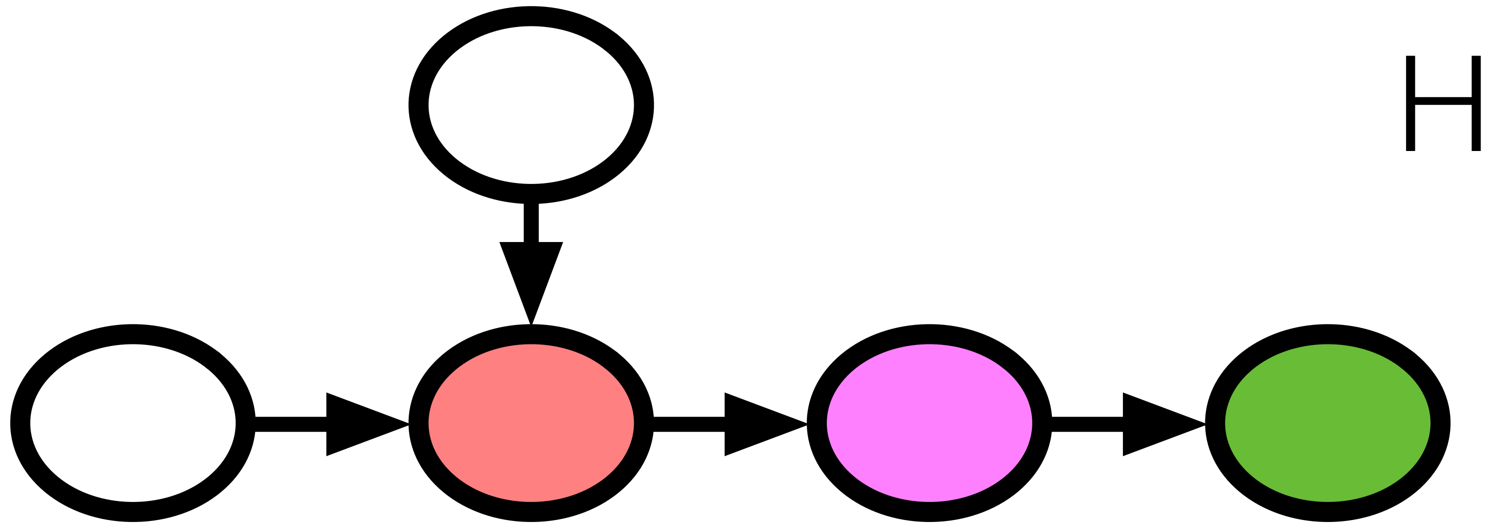


Frameworks

Hardware

Existing Deep Learning Frameworks

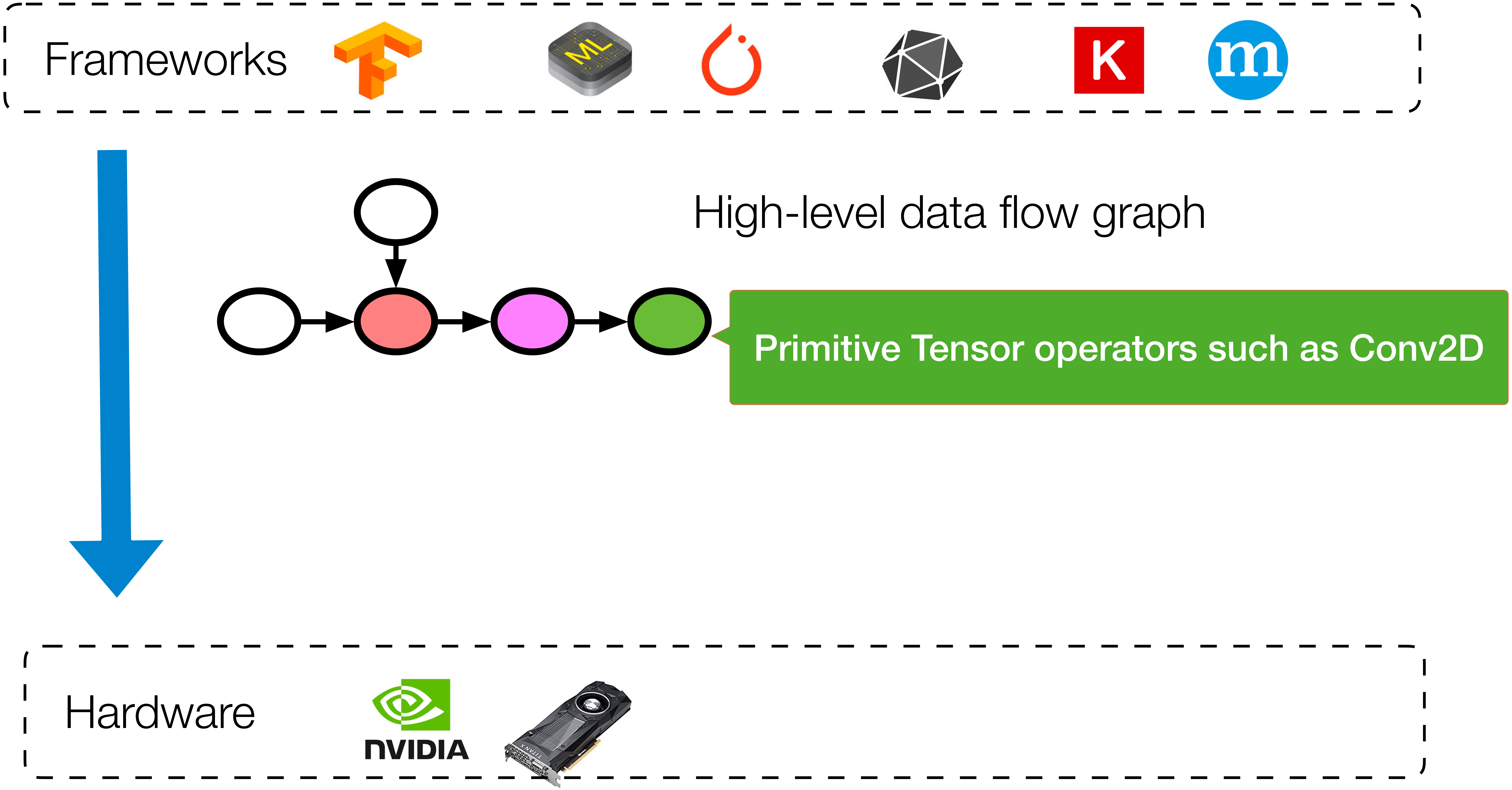
Frameworks



High-level data flow graph

Hardware

Existing Deep Learning Frameworks



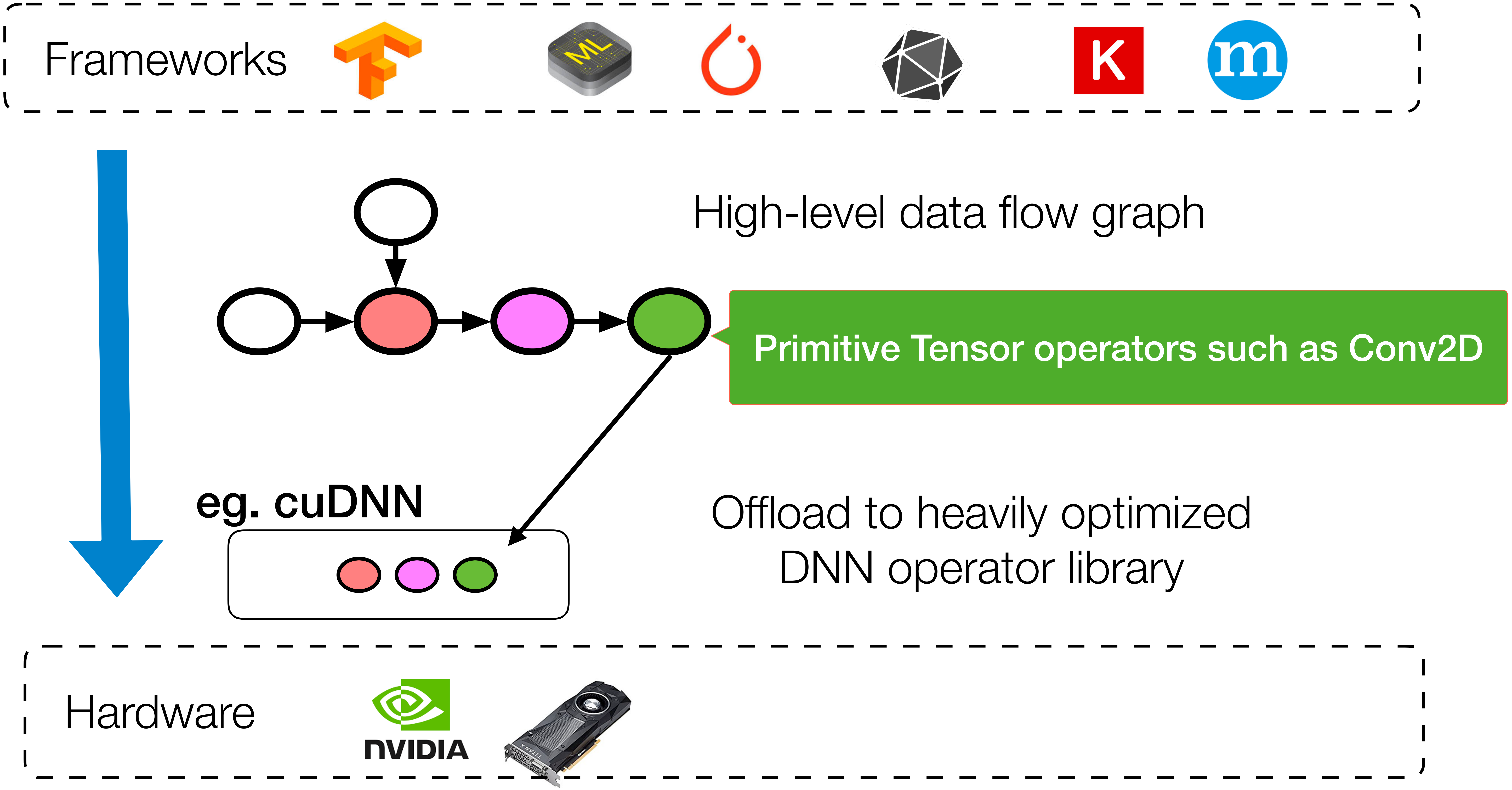
Frameworks

High-level data flow graph

Primitive Tensor operators such as Conv2D

Hardware

Existing Deep Learning Frameworks



Frameworks

High-level data flow graph

Primitive Tensor operators such as Conv2D

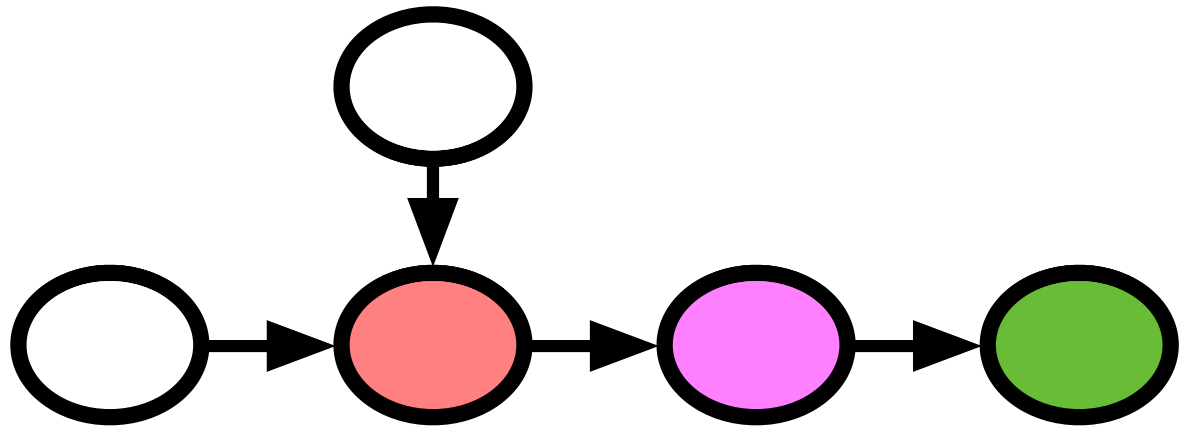
eg. cuDNN

Offload to heavily optimized DNN operator library

Hardware

Limitations of Existing Approach

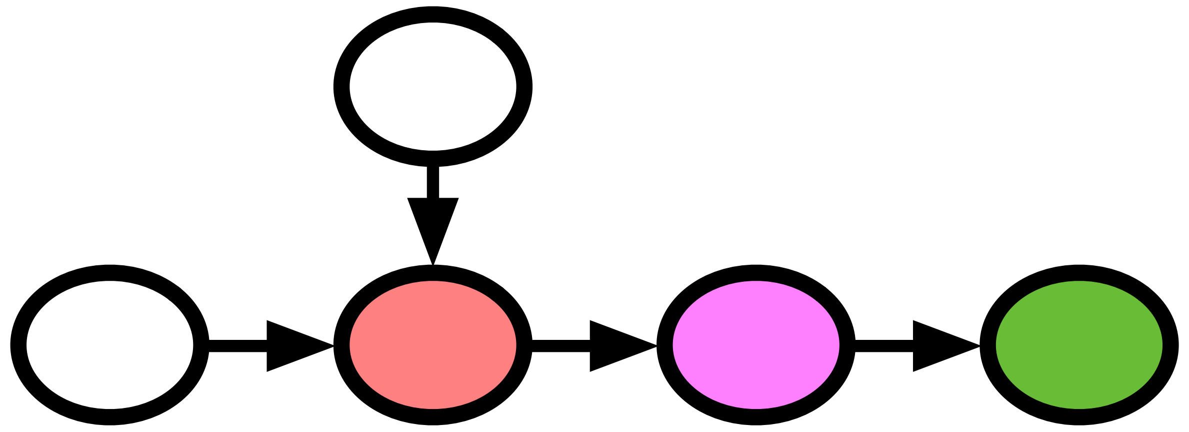
Frameworks



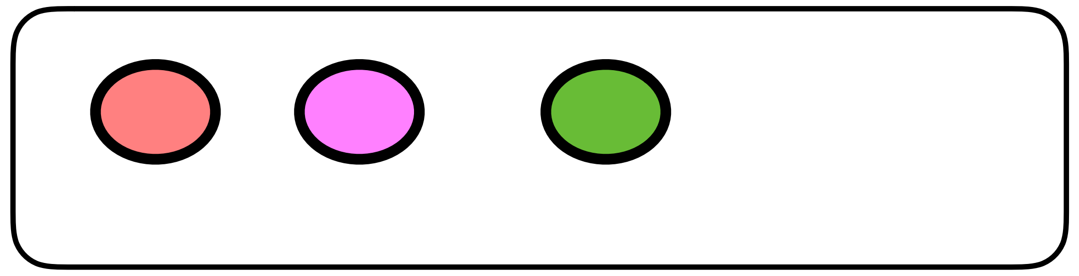
cuDNN

Limitations of Existing Approach

Frameworks

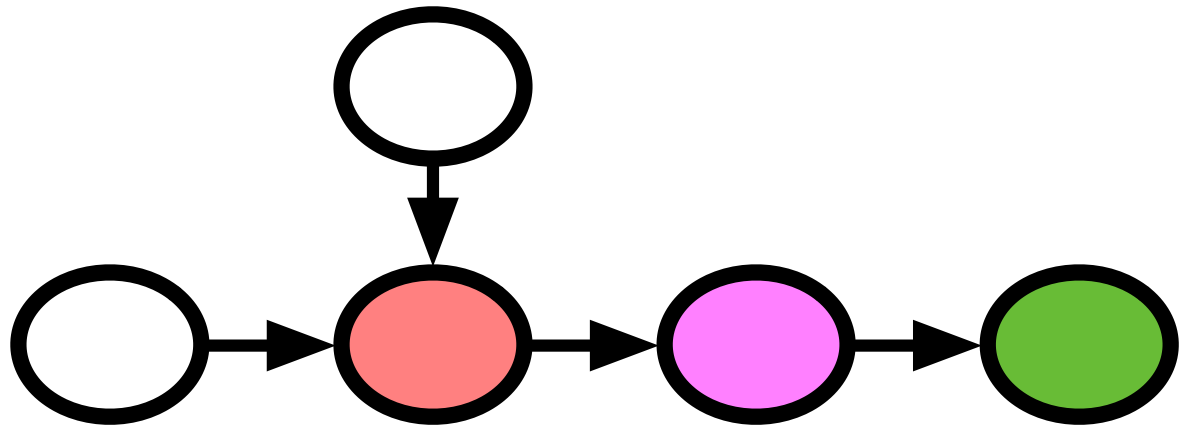


cuDNN

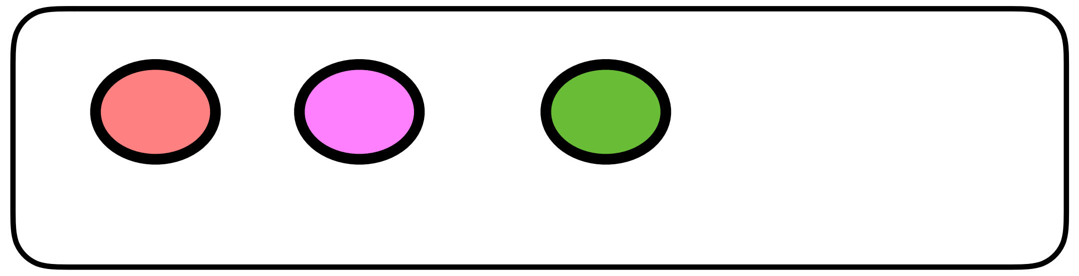


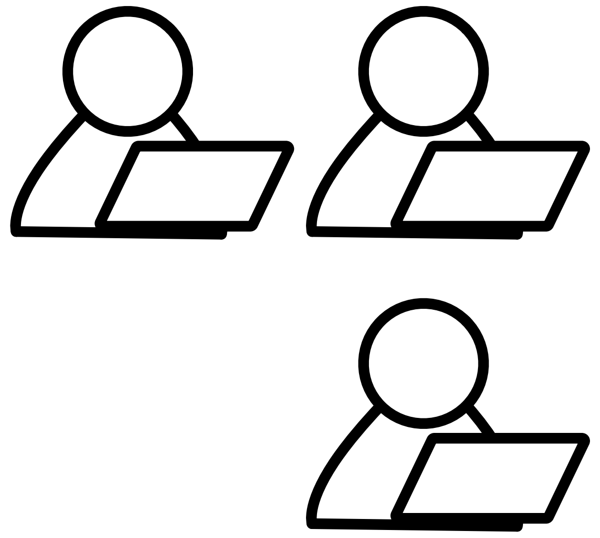
Limitations of Existing Approach

Frameworks



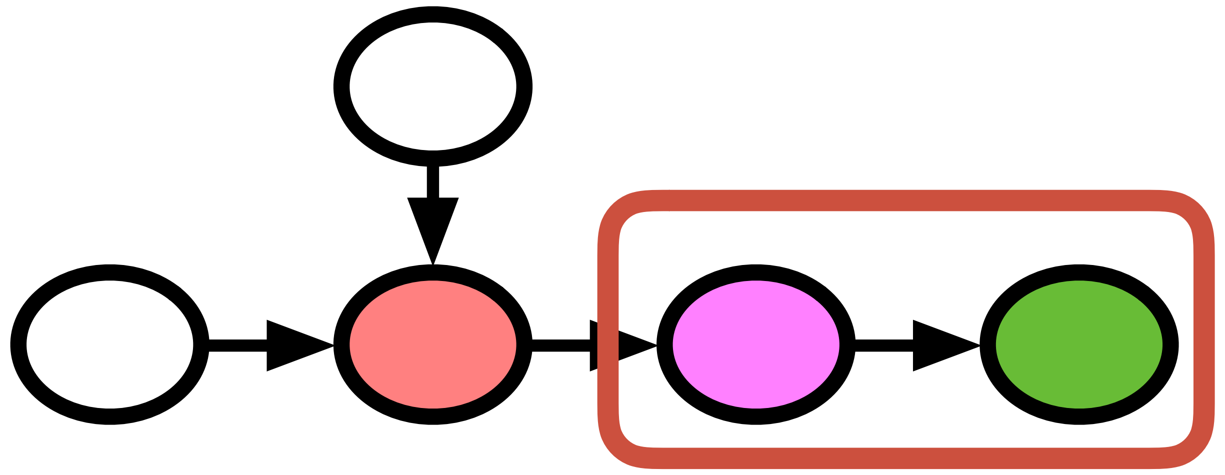
cuDNN



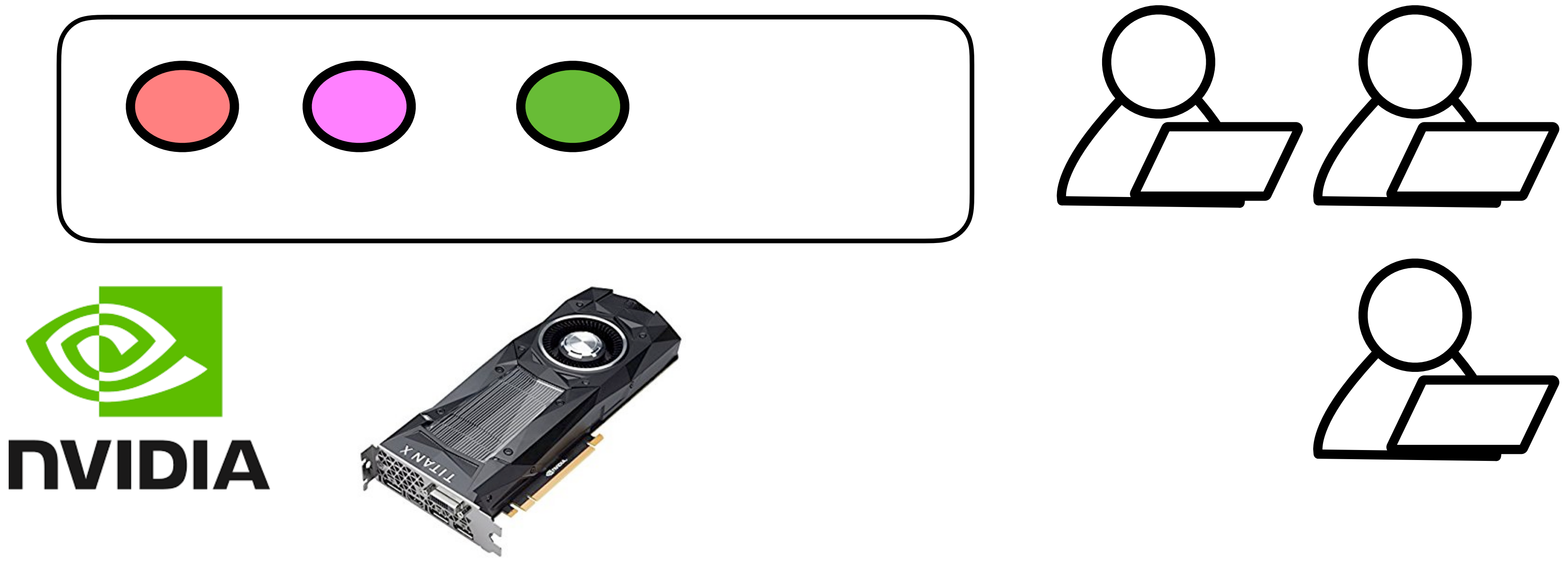


Limitations of Existing Approach

Frameworks

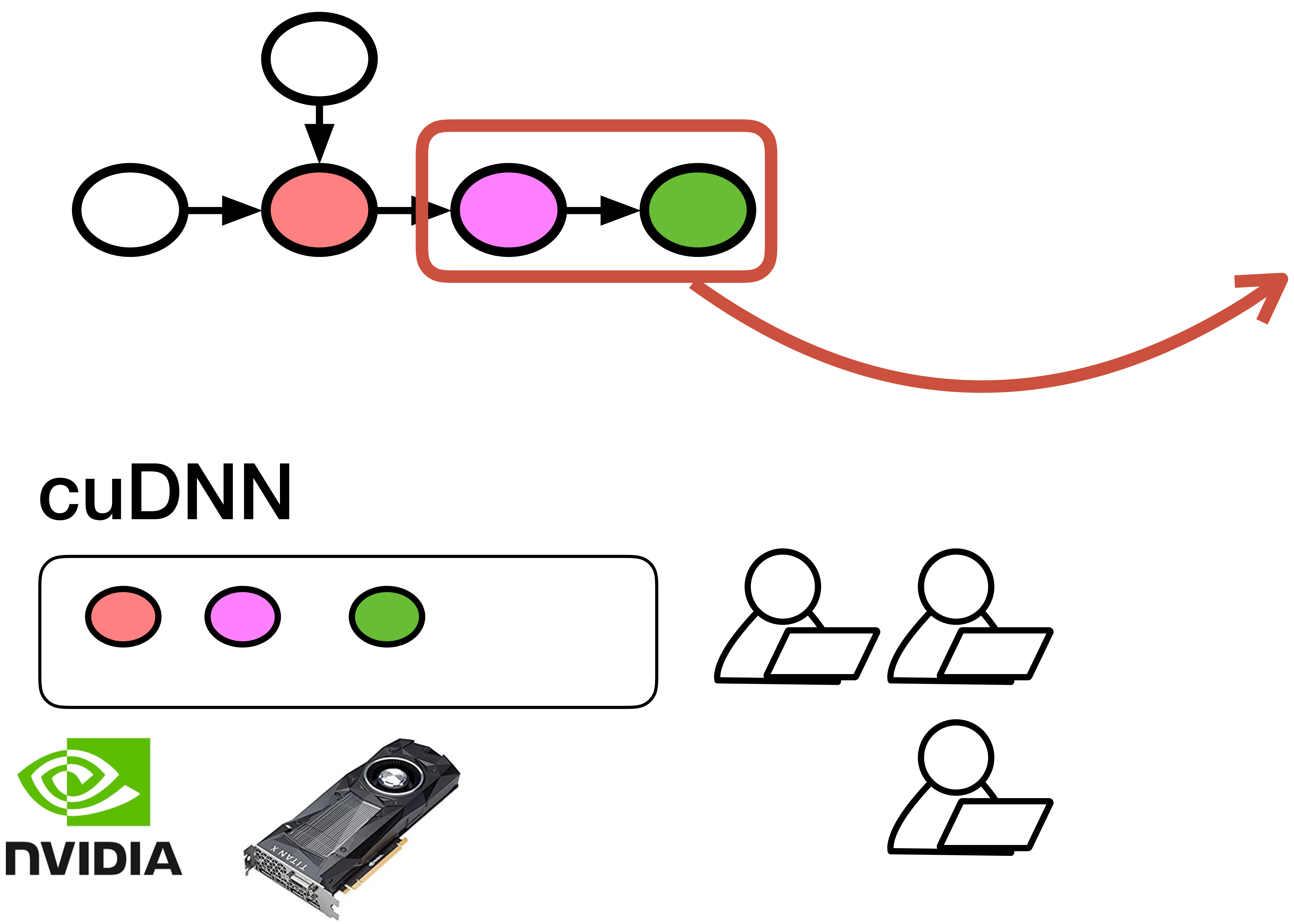


cuDNN



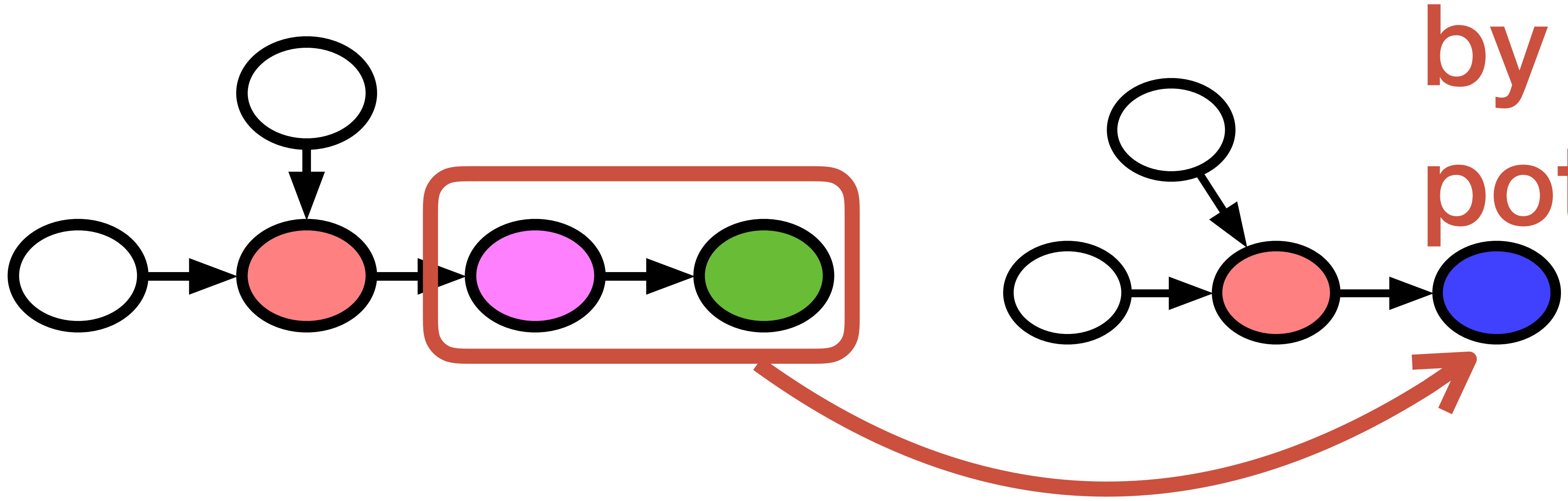
Limitations of Existing Approach

Frameworks



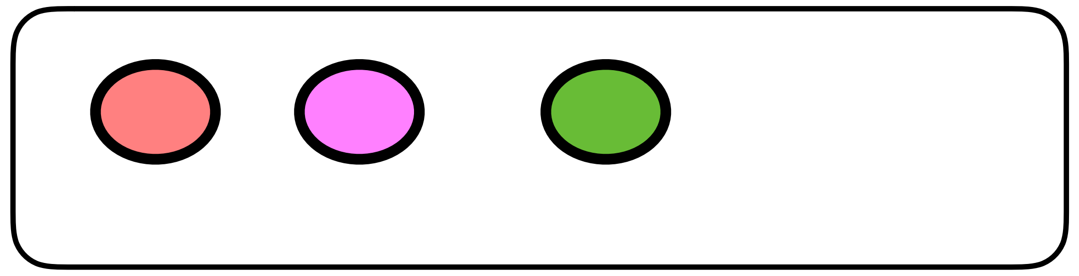
Limitations of Existing Approach

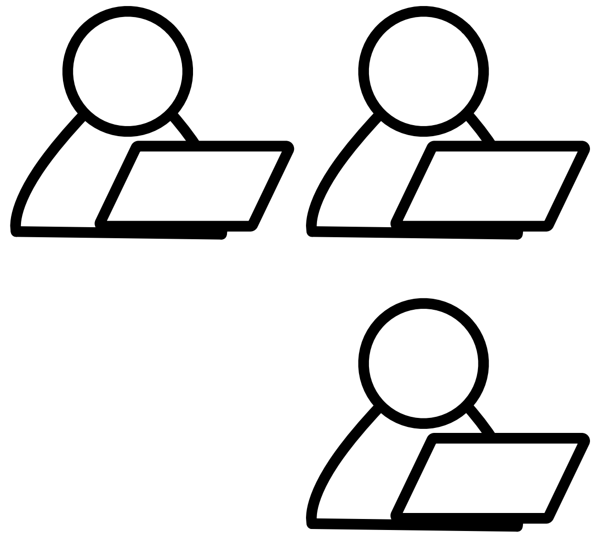
Frameworks



New operator introduced
by operator fusion optimization
potential benefit: 1.5x speedup

cuDNN

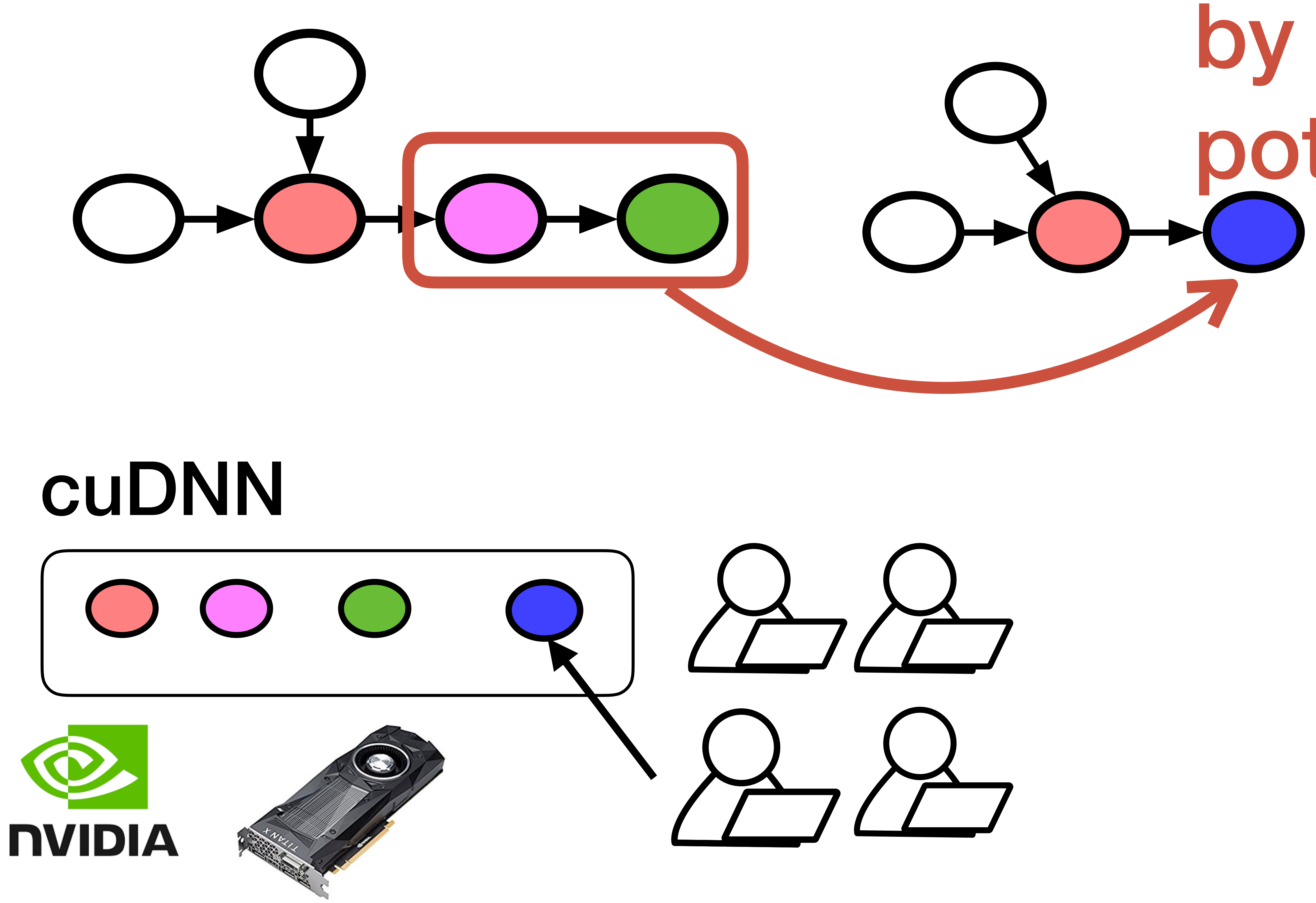




Limitations of Existing Approach

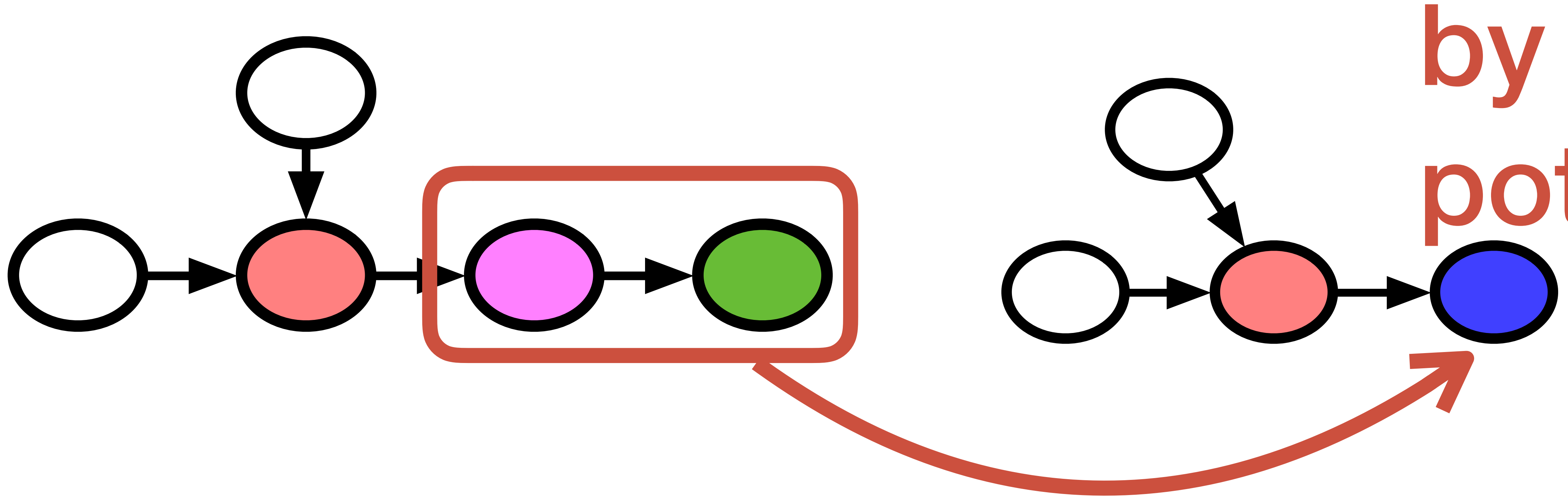
Frameworks

New operator introduced
by operator fusion optimization
potential benefit: 1.5x speedup

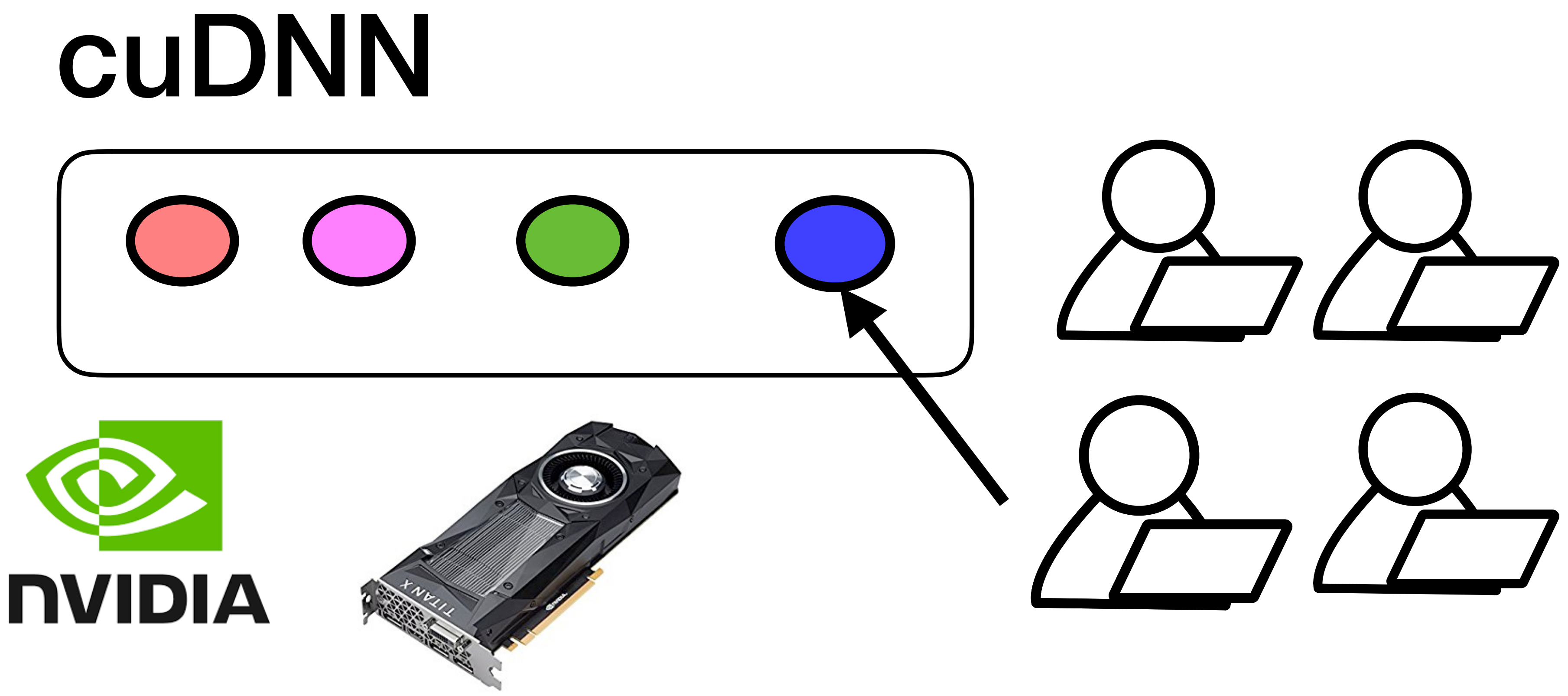


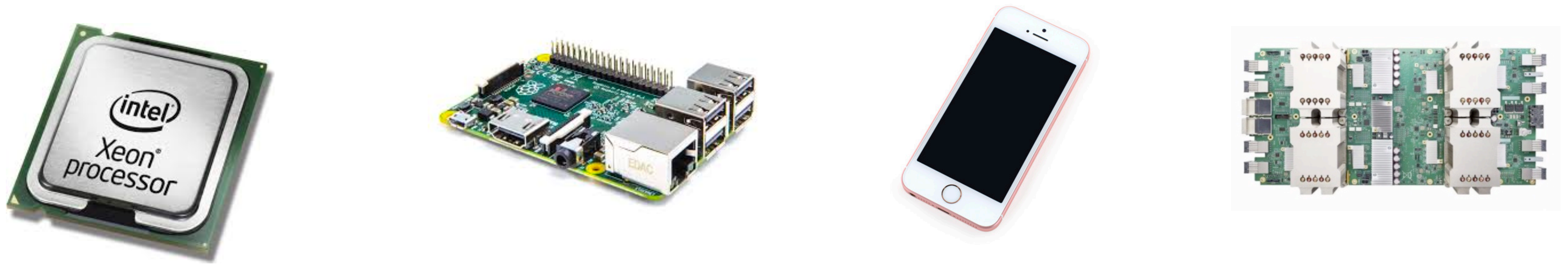
Limitations of Existing Approach

Frameworks



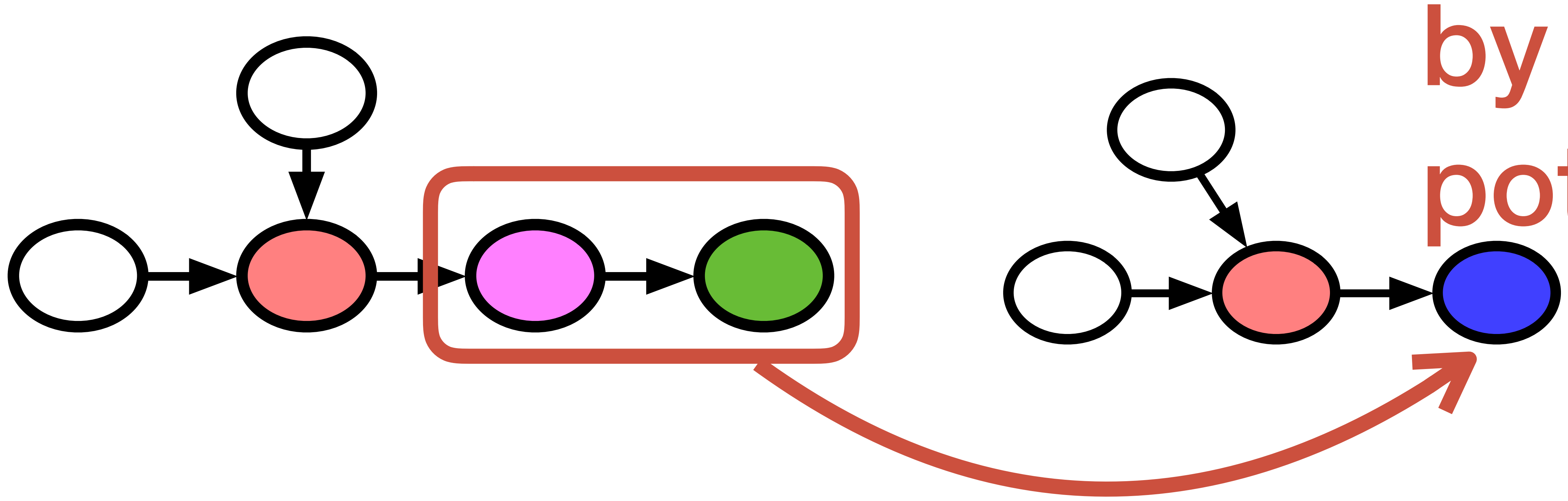
New operator introduced
by operator fusion optimization
potential benefit: 1.5x speedup





Limitations of Existing Approach

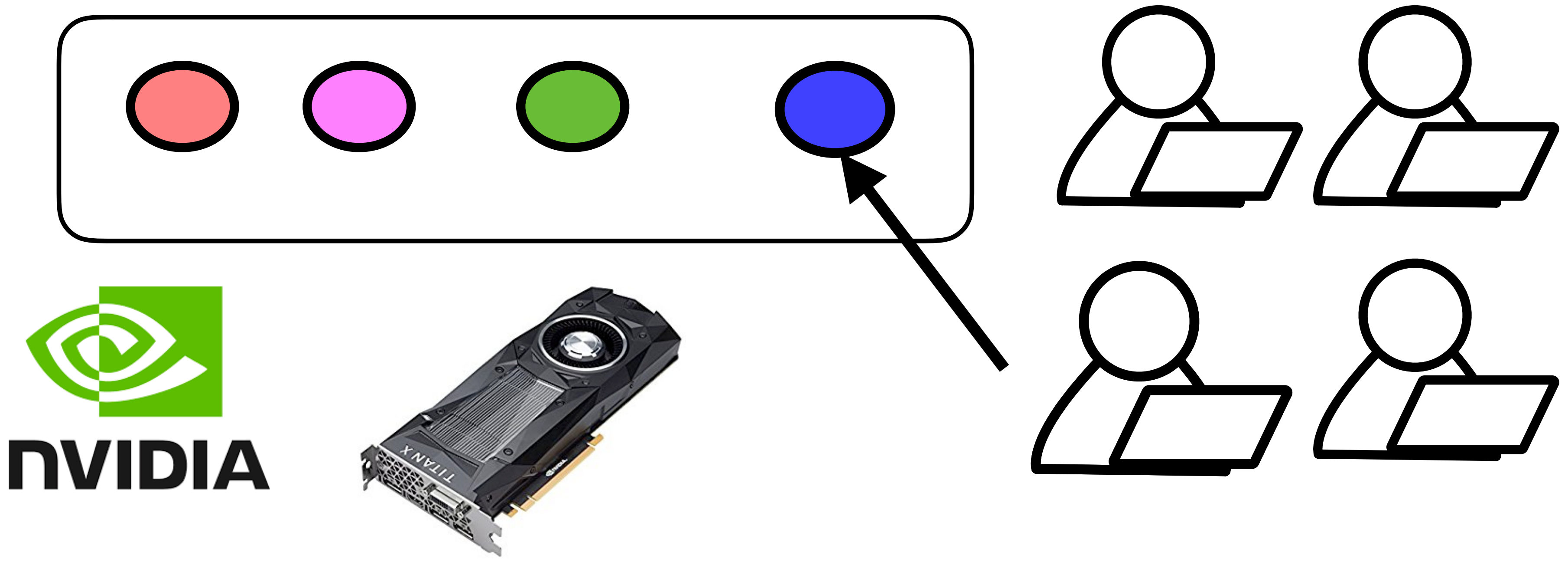
Frameworks

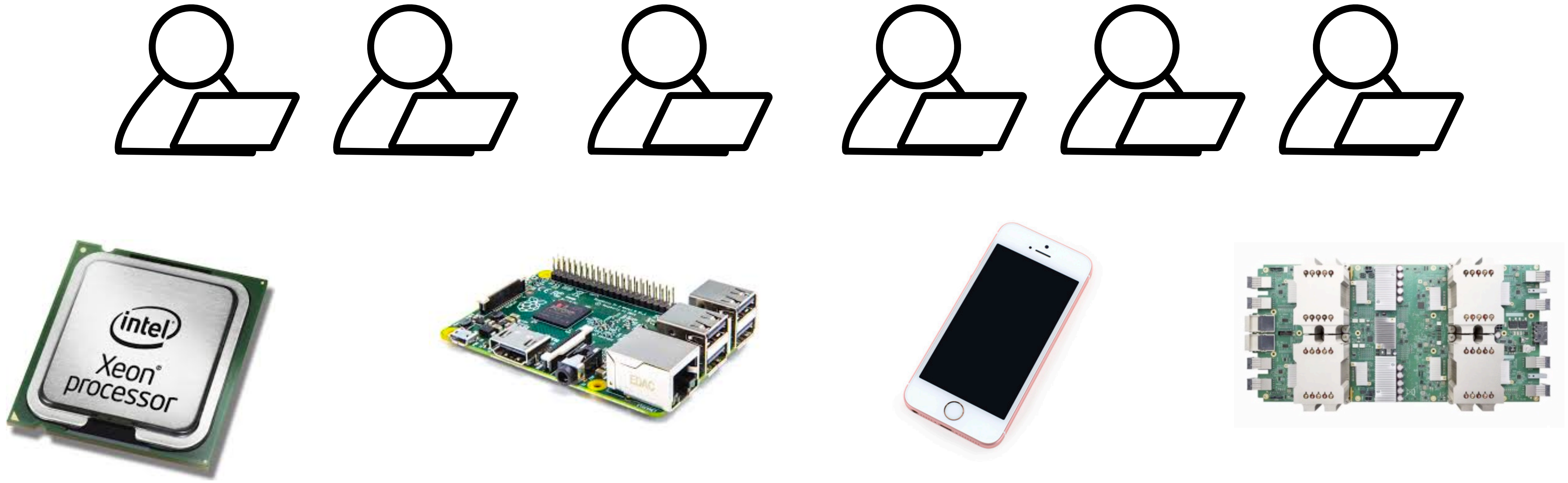


New operator introduced
by operator fusion optimization
potential benefit: 1.5x speedup

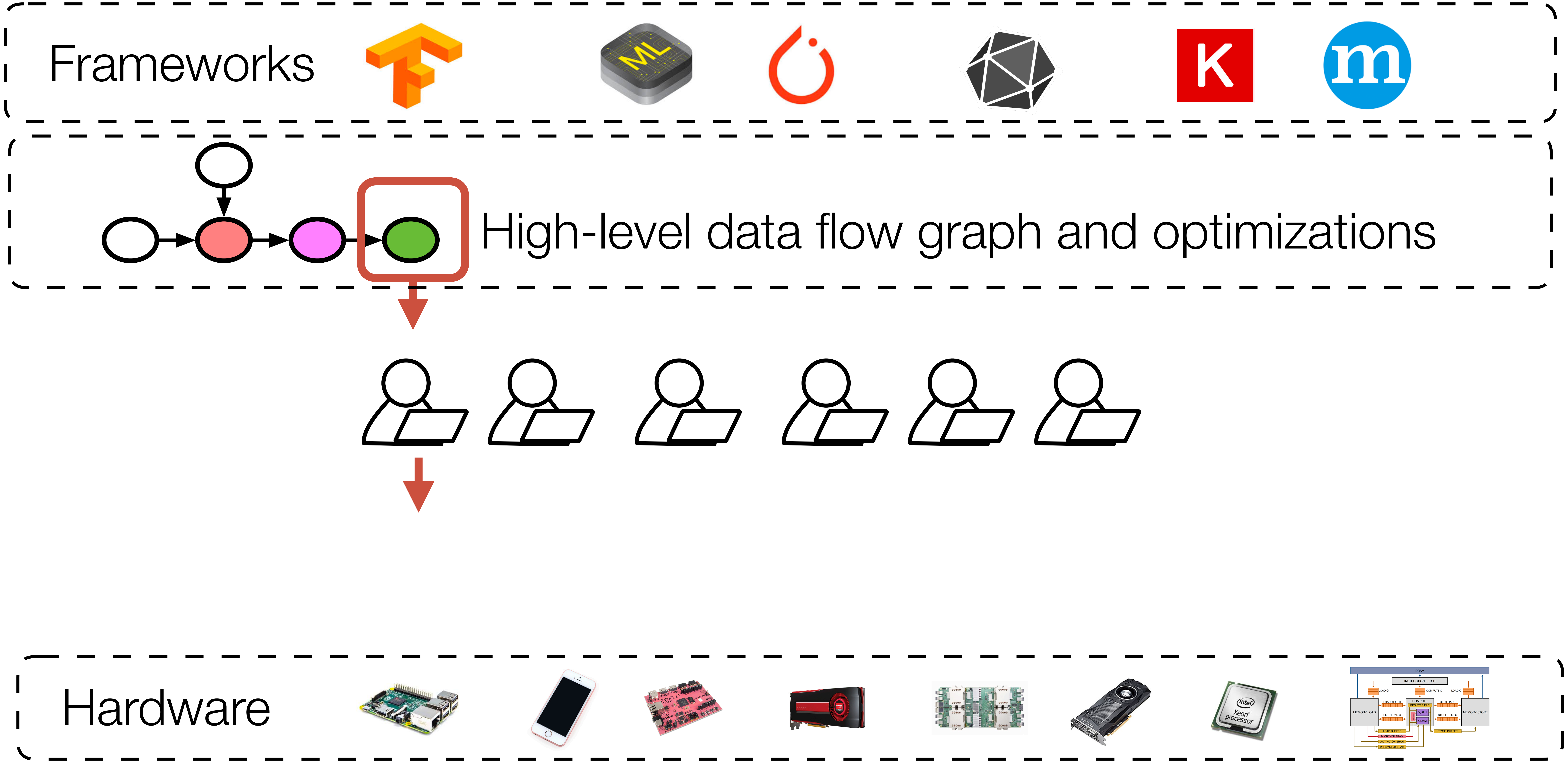
Engineering intensive

cuDNN





TVM: Learning-based Learning System

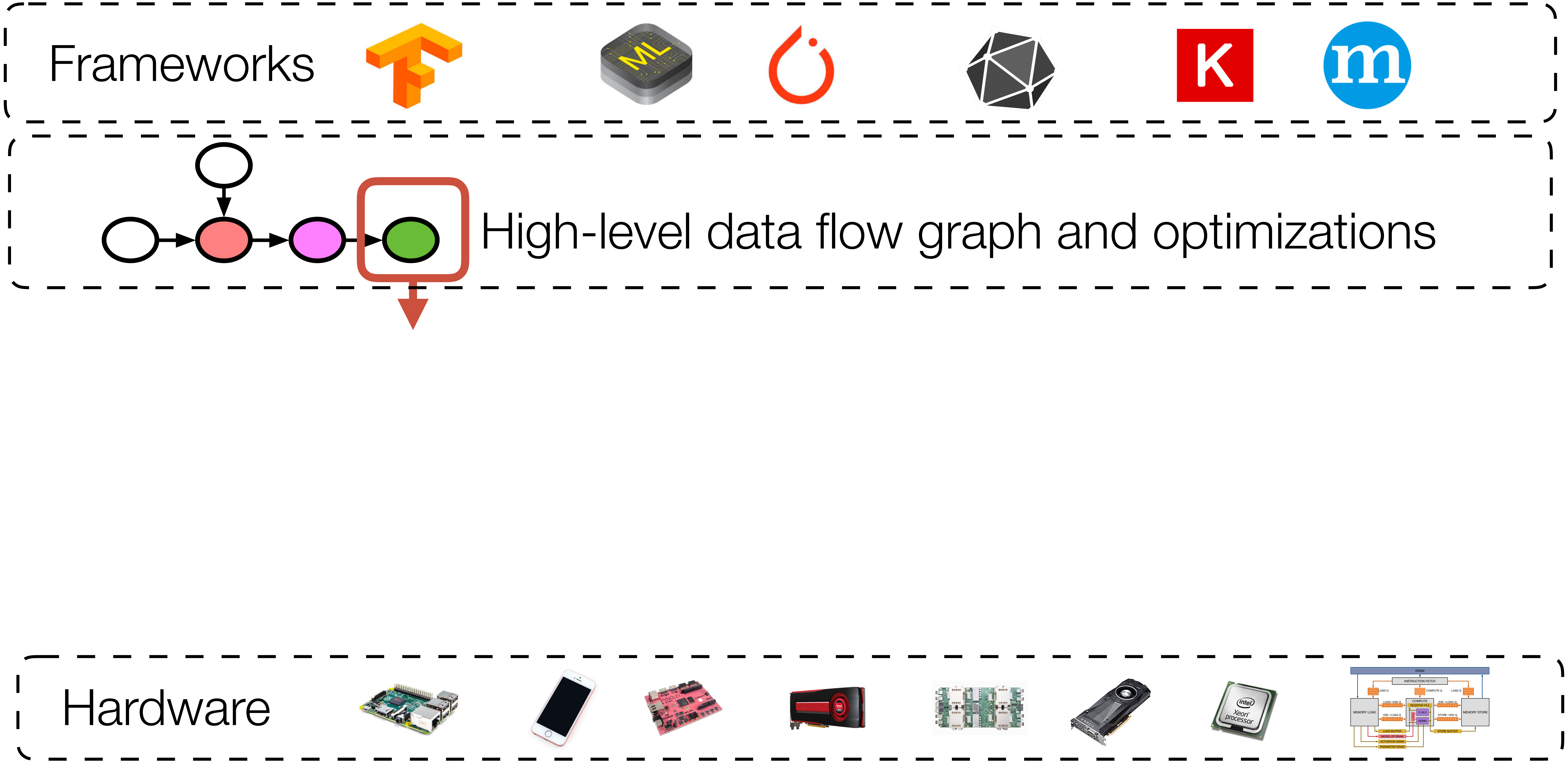


Frameworks

High-level data flow graph and optimizations

Hardware

TVM: Learning-based Learning System

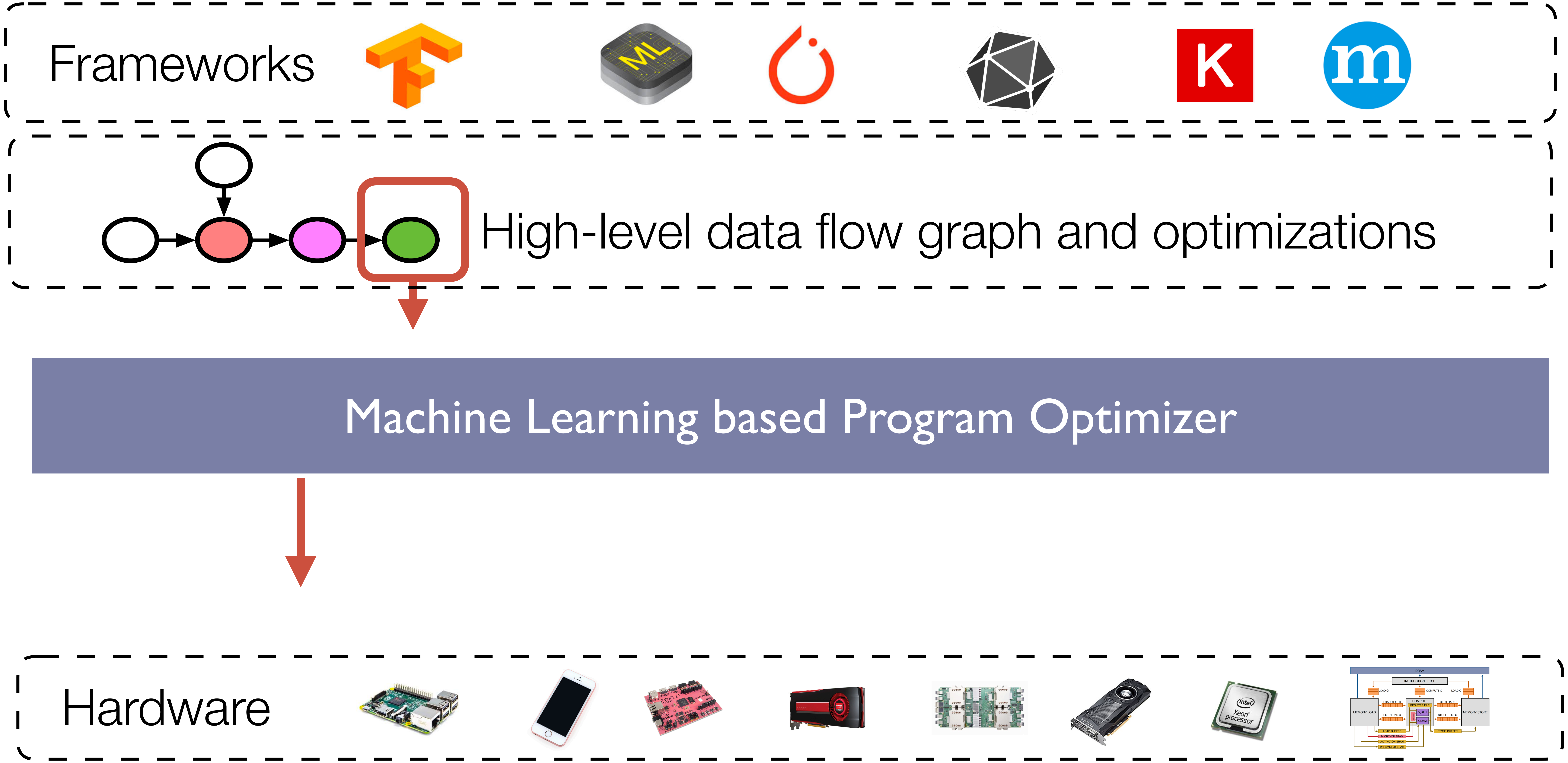


Frameworks

High-level data flow graph and optimizations

Hardware

TVM: Learning-based Learning System



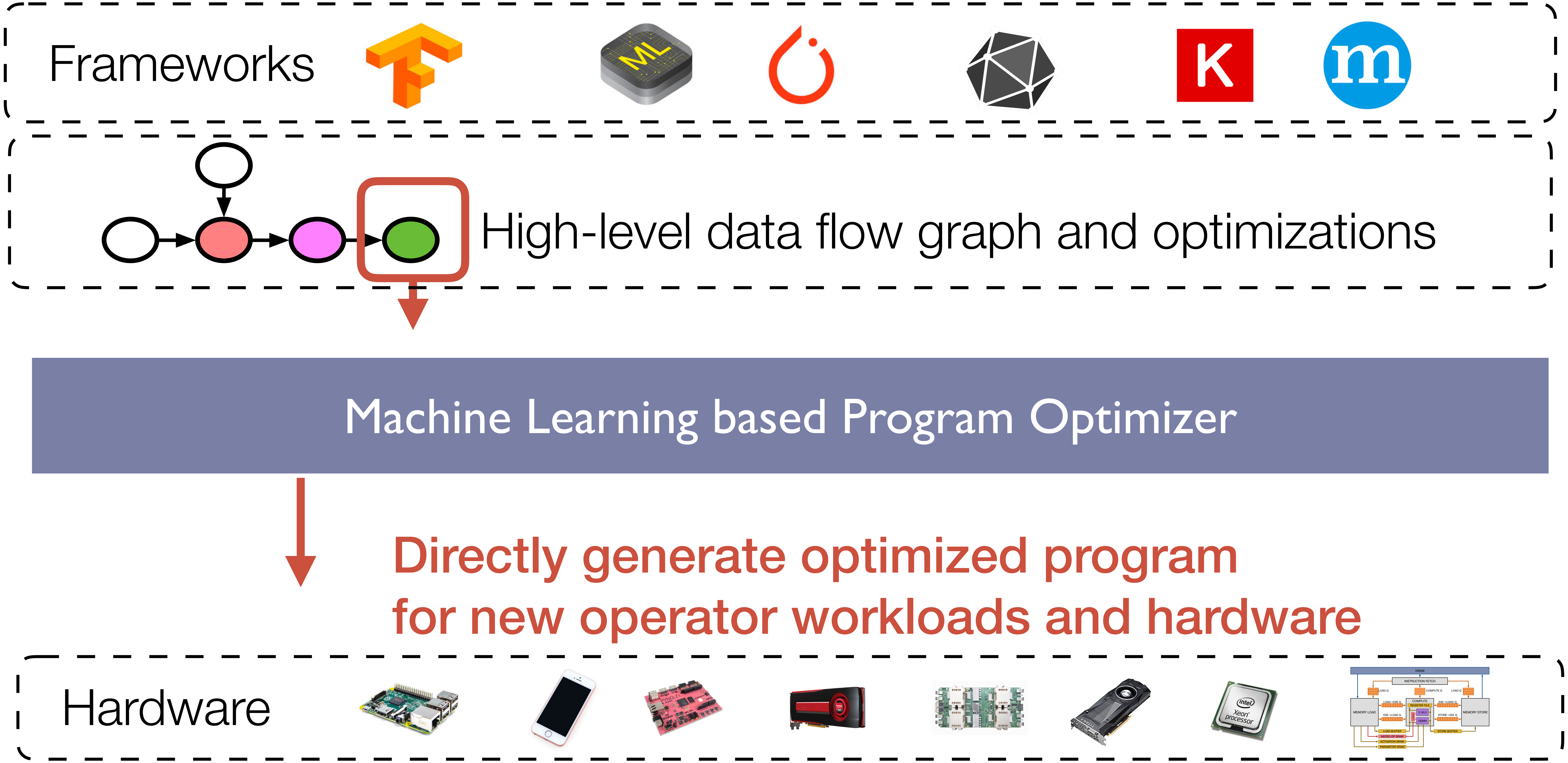
Frameworks

High-level data flow graph and optimizations

Machine Learning based Program Optimizer

Hardware

TVM: Learning-based Learning System



Frameworks

High-level data flow graph and optimizations

Machine Learning based Program Optimizer

Directly generate optimized program for new operator workloads and hardware

Hardware

Why Automation is the Future

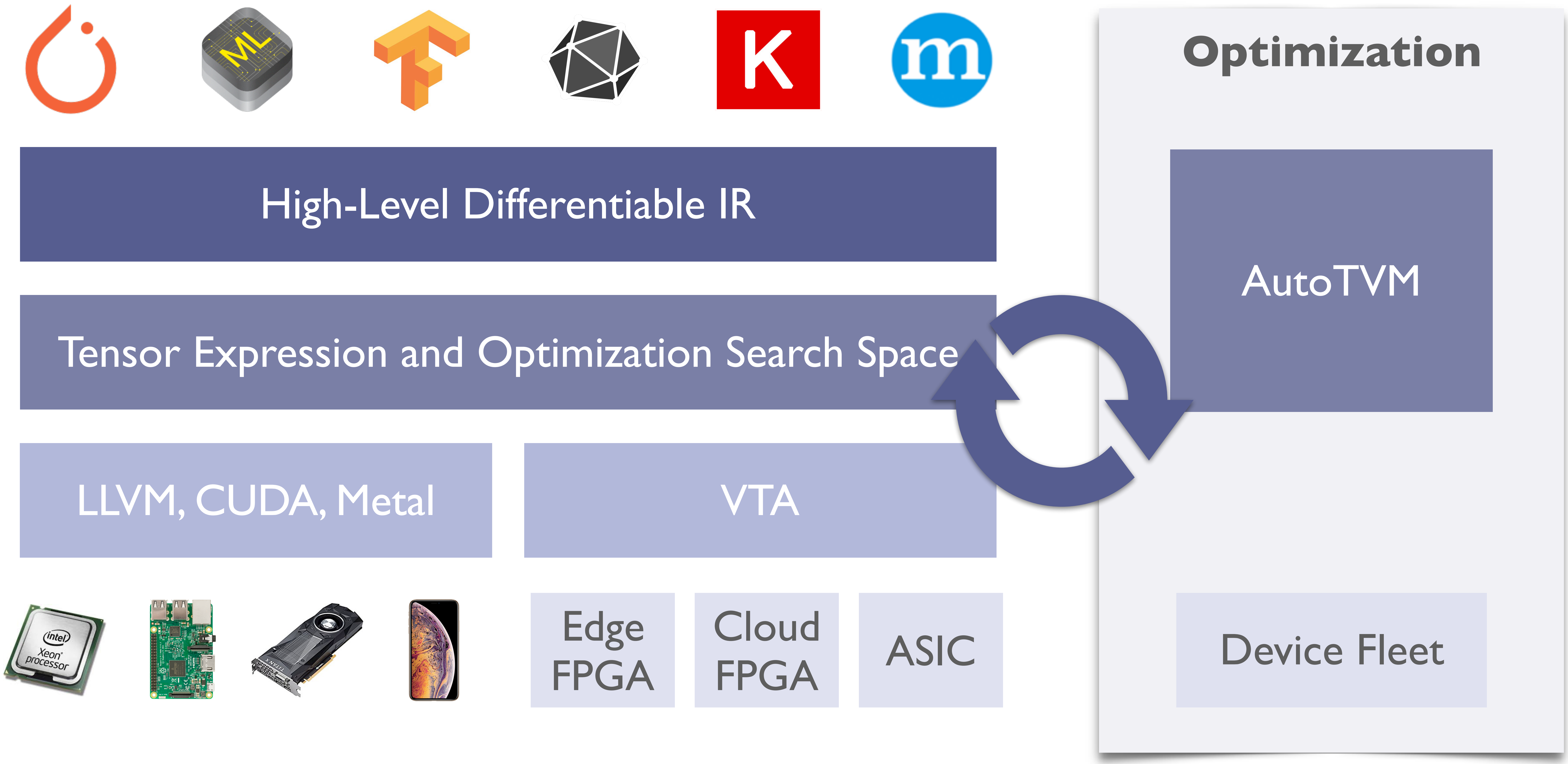
Clear winner on emerging models in product

Competitive on benchmarking type model

Quickly enables other optimizations: fusion, layout, parallelization

Portable performance across devices

TVM Stack



High-Level Differentiable IR

Tensor Expression and Optimization Search Space

LLVM, CUDA, Metal

VTA

Edge FPGA

Cloud FPGA

ASIC

Optimization

AutoTVM

Device Fleet

Community Highlights

More **Dynamism**

Tiny machine learning

Better core **Infra**

More Specialized **Accelerator Support**

Community Highlights

More **Dynamism**

Tiny machine learning

Better core **Infra**

More Specialized **Accelerator Support**

Need for More Dynamism

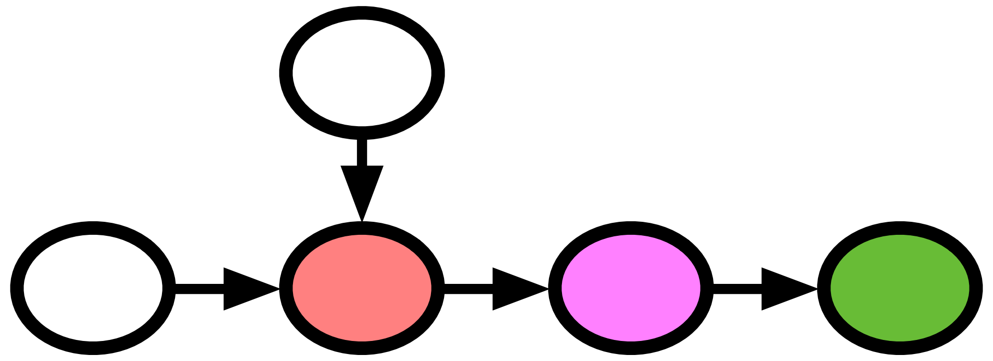
Model

Data

Need for More Dynamism

static
computational graph

Model

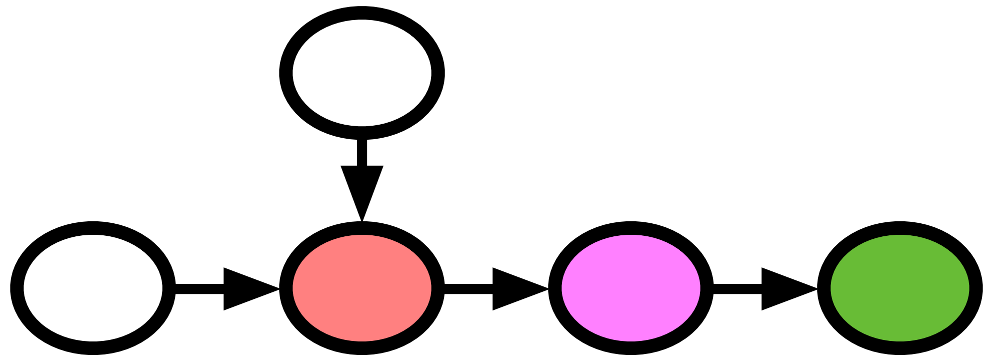


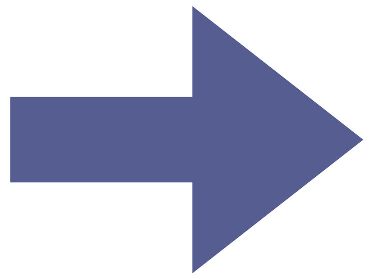
Data

Need for More Dynamism

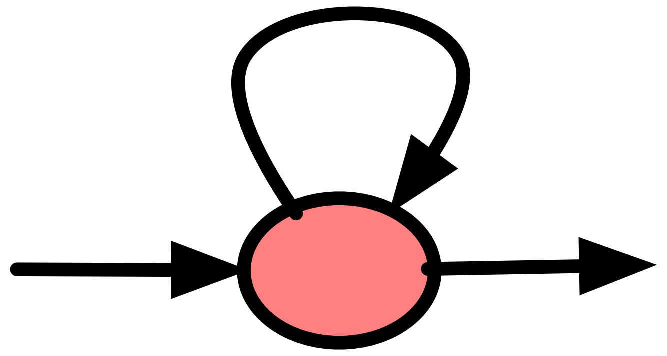
Model

static
computational graph





program with
loops and recursions

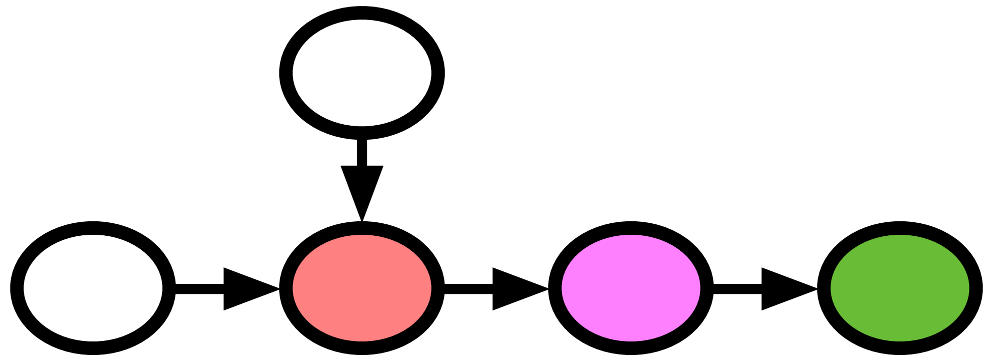


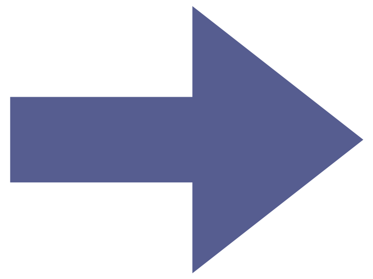
Data

Need for More Dynamism

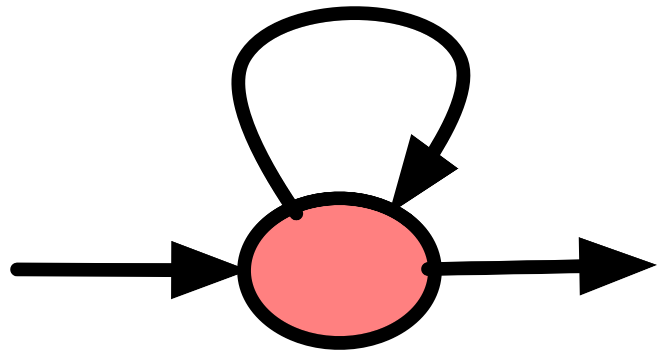
Model

static
computational graph





program with
loops and recursions



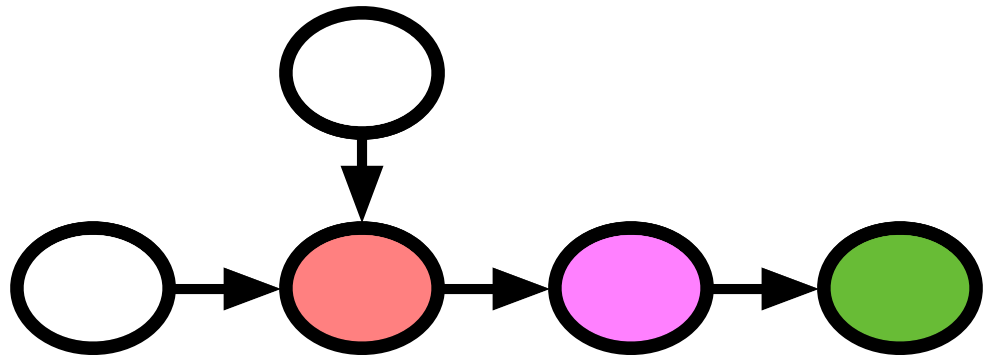
Data

single tensor
with known shape

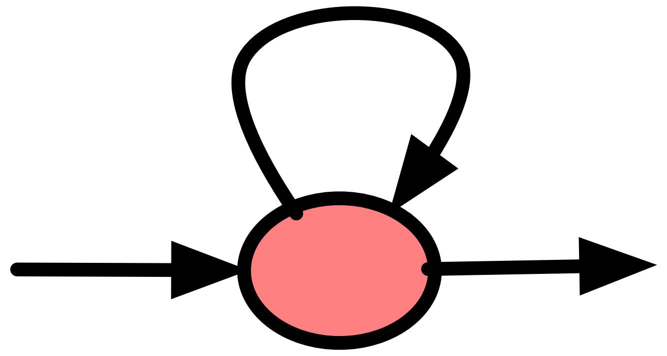
Need for More Dynamism

Model

static
computational graph



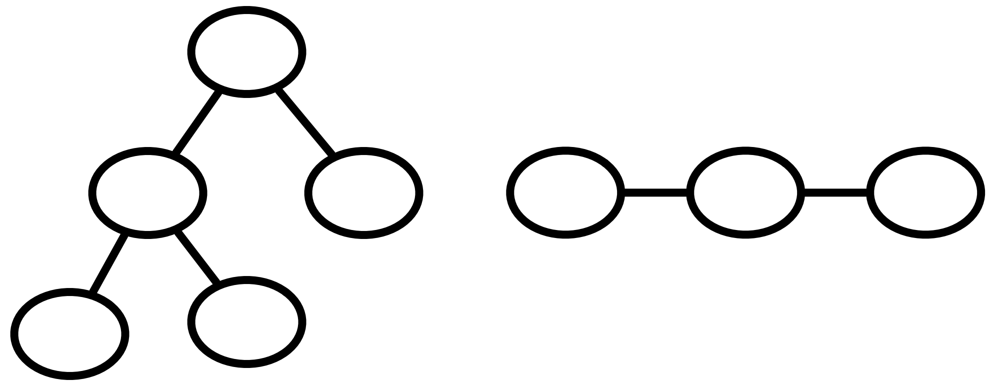
program with
loops and recursions



Data

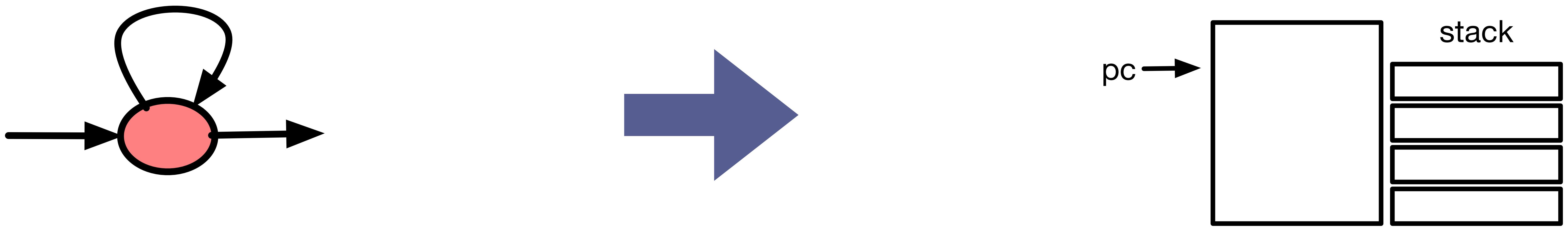
single tensor
with known shape

sequence, trees,
nested data structure



Relay Virtual Machine

source program **VM bytecode and runtime**



Dynamic shape workloads

More runtime objects: Arrays, Tuples, Trees, ADTs

Minimum runtime for dynamic models

Credit: Jared Roesch, Haichen Shen et.al

Community Highlights

More **Dynamism**

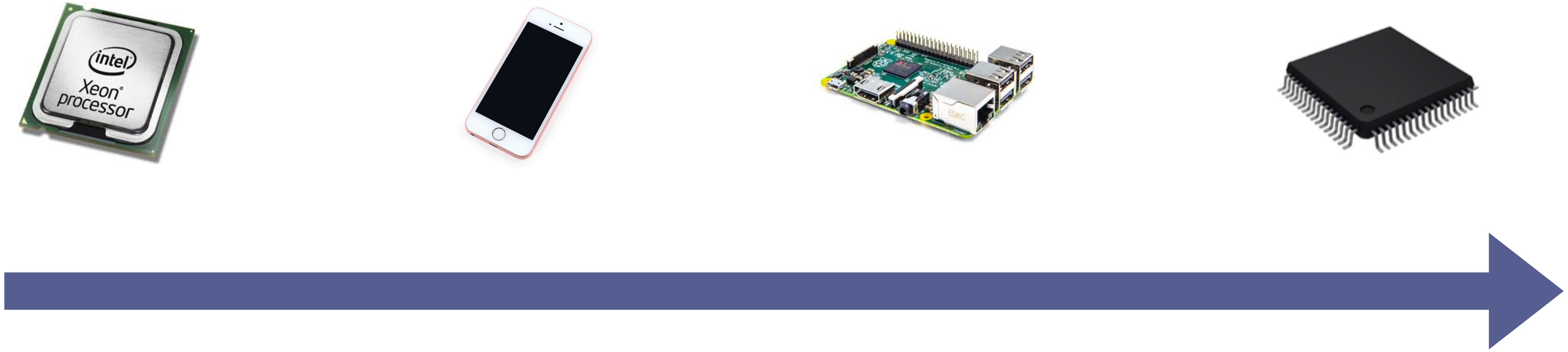
Tiny machine learning

Better core **Infra**

More Specialized **Accelerator Support**

Machine Learning is Getting into Tiny Devices

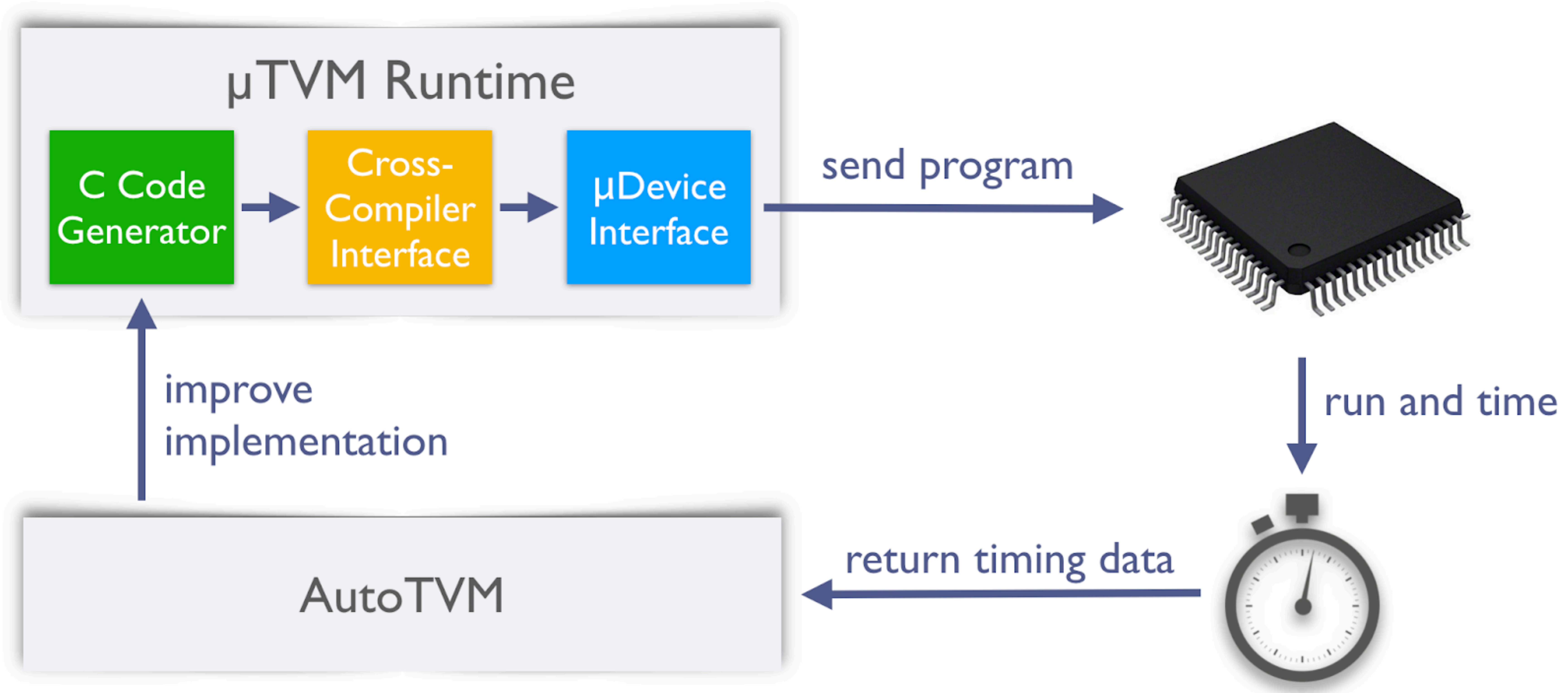
Challenges: limited resources, OS support



uTVM: TVM on bare-metal Devices

Support bare-metal J-TAG devices, **no OS is needed**

ARM Cortex-M
RISC-V



Credit: Logan Weber et al

Community Highlights

More **Dynamism**

Tiny machine learning

Better core **Infra**

More Specialized **Accelerator Support**

Core Infrastructure

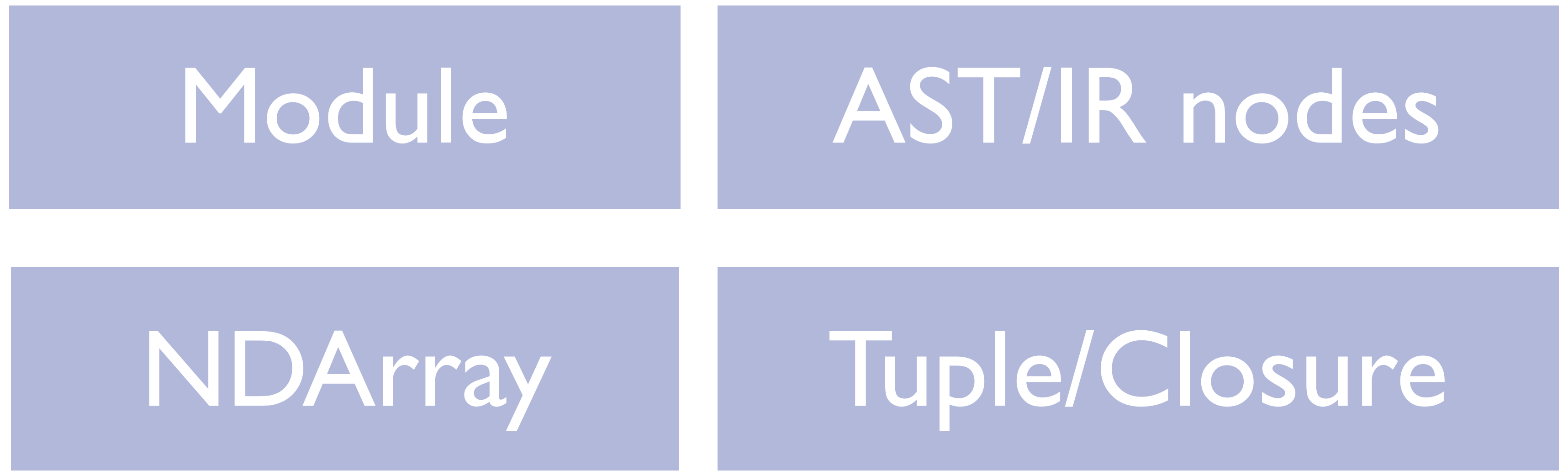
New integer simplification and analysis

Unified runtime object protocol

Core Infrastructure

New integer simplification and analysis

Unified runtime object protocol



Module

AST/IR nodes

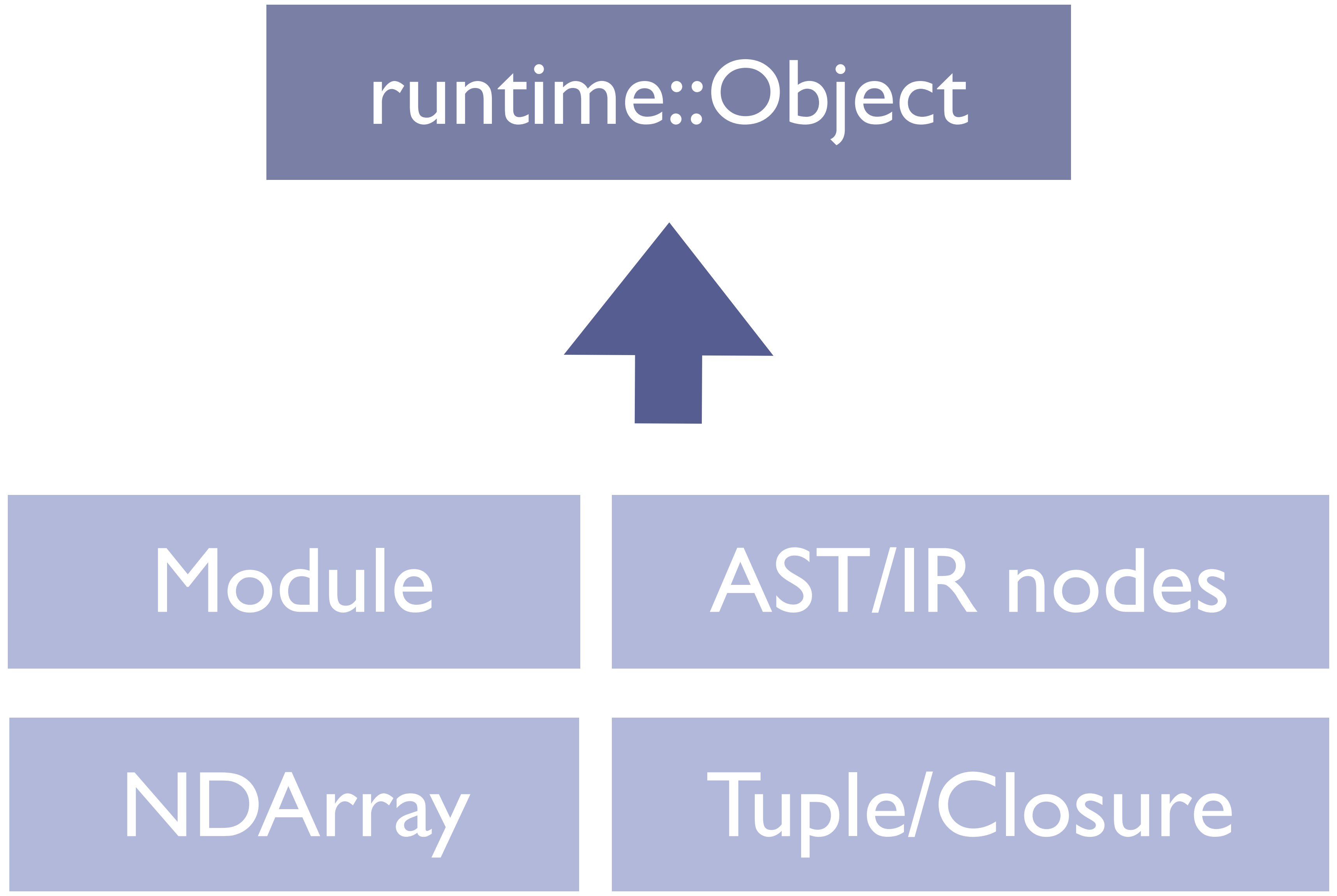
NDArray

Tuple/Closure

Core Infrastructure

New integer simplification and analysis

Unified runtime object protocol



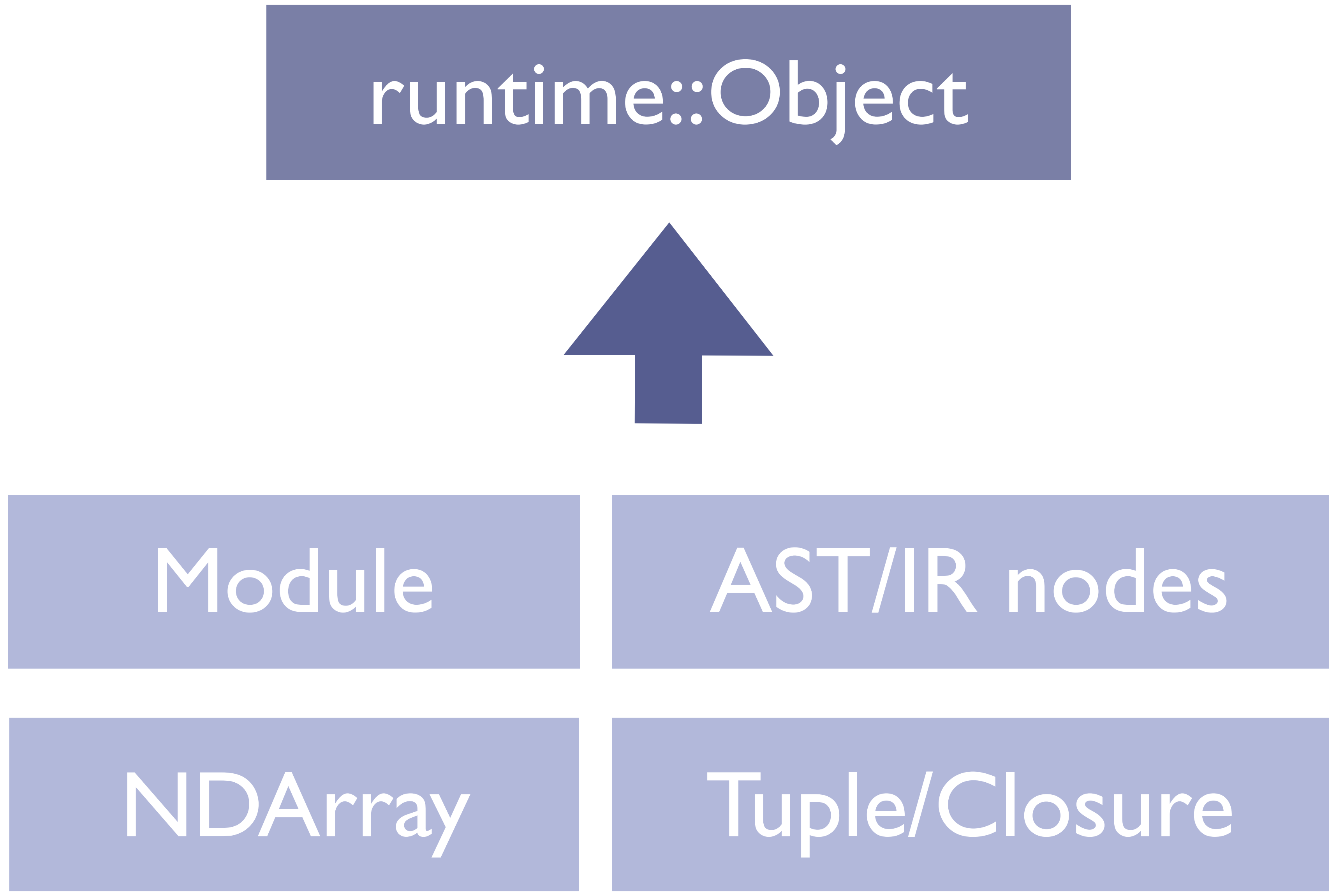
Core Infrastructure

New integer simplification and analysis

Unified runtime object protocol

Easy to add new objects (trees, graphs)

Cross language support



Community Highlights

More **Dynamism**

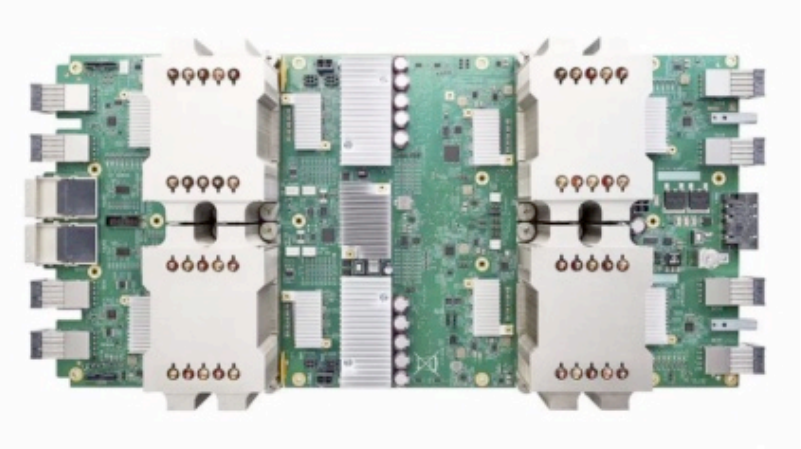
Tiny machine learning

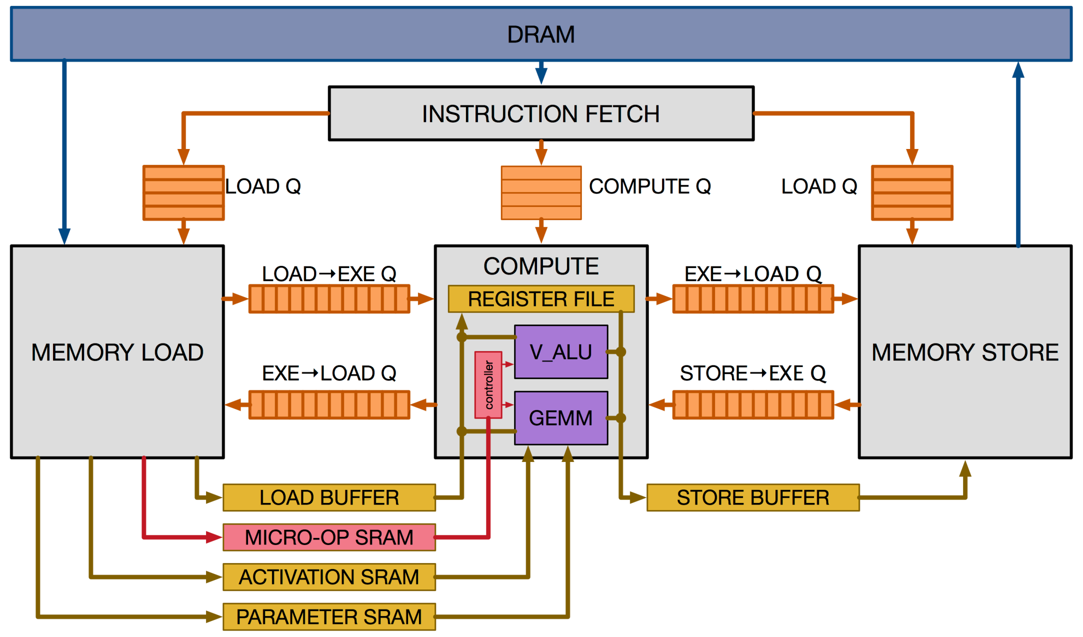
Better core **Infra**

More Specialized **Accelerator Support**

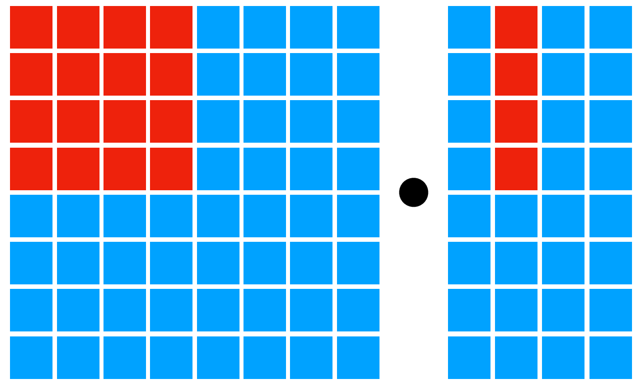
Tensorization Challenge for Specialized Accelerators

TPUs

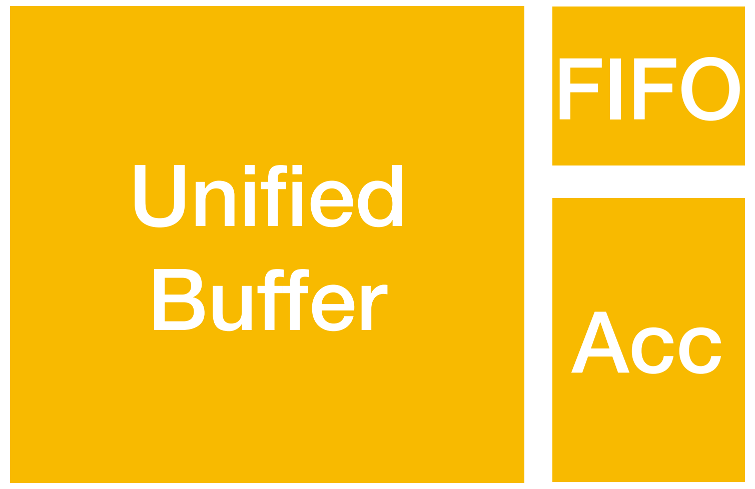




Tensor Compute Primitives



Explicitly Managed Memory Subsystem



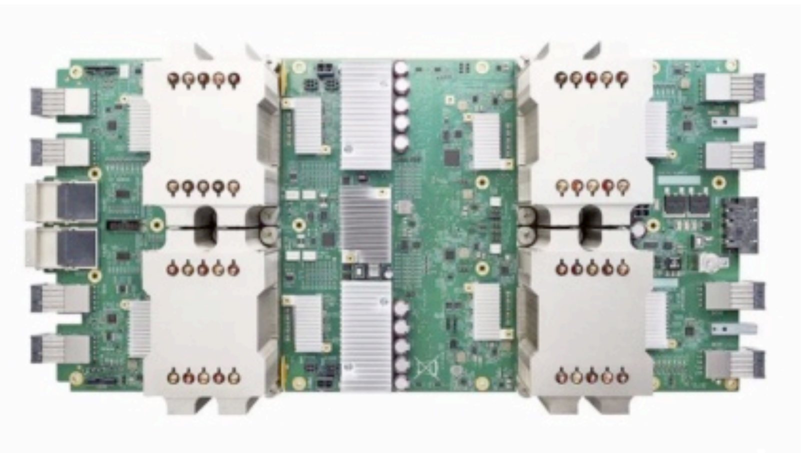
Unified Buffer

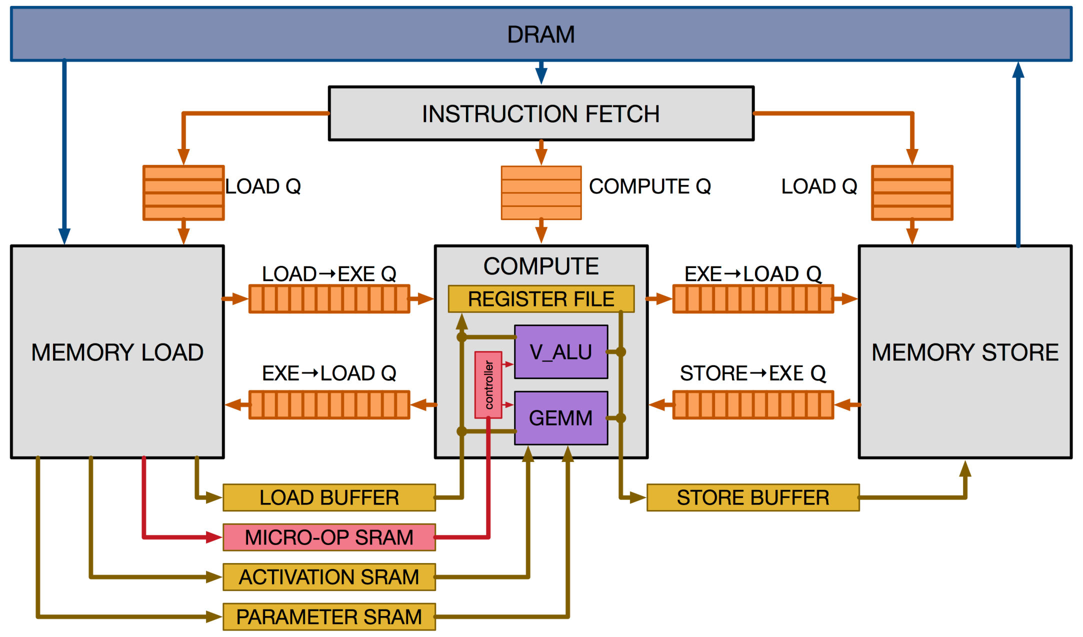
FIFO

Acc

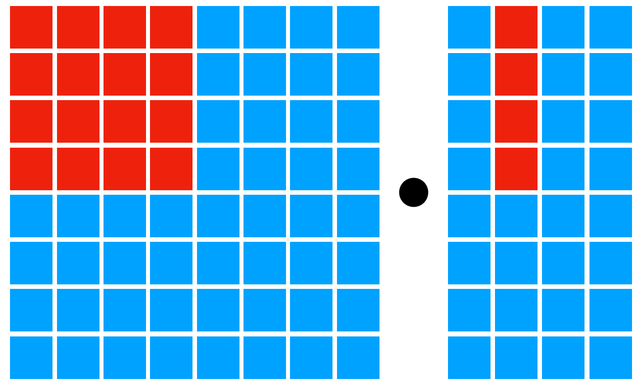
Tensorization Challenge for Specialized Accelerators

TPUs

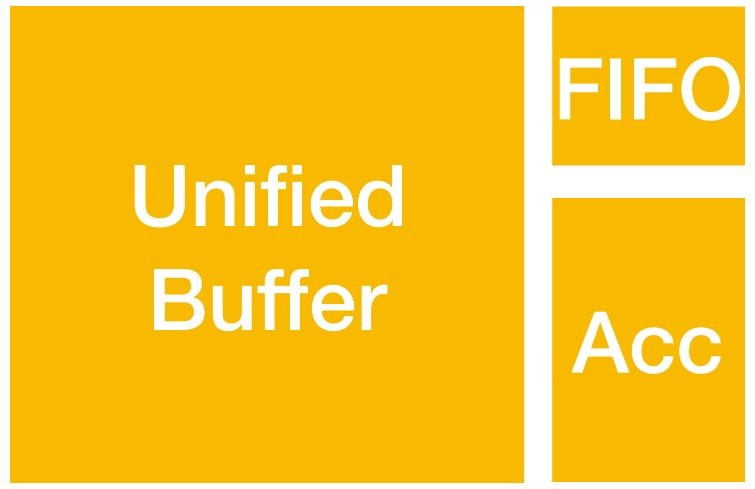




Tensor Compute Primitives



Explicitly Managed Memory Subsystem

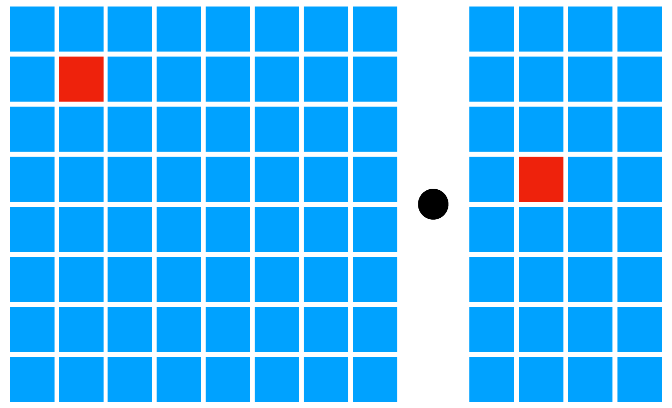


Tensorization Challenge

Compute primitives

Tensorization Challenge

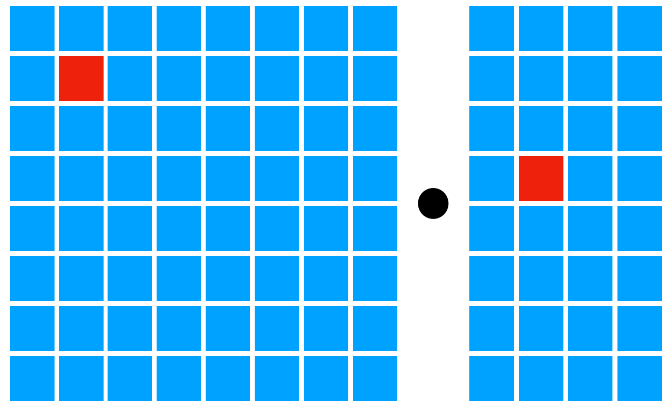
Compute primitives



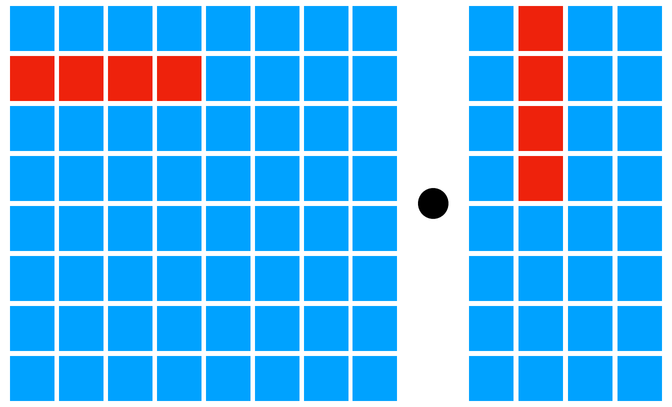
scalar

Tensorization Challenge

Compute primitives



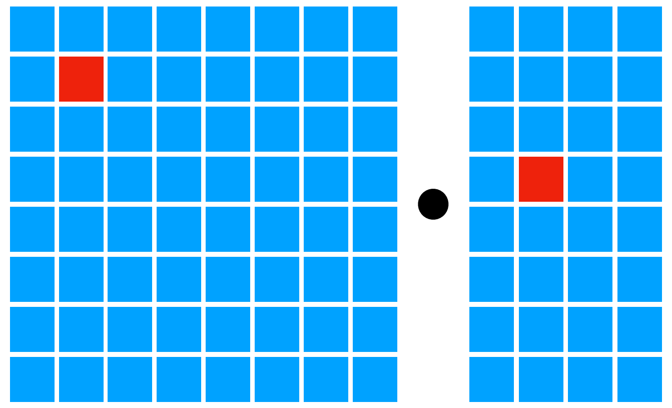
scalar



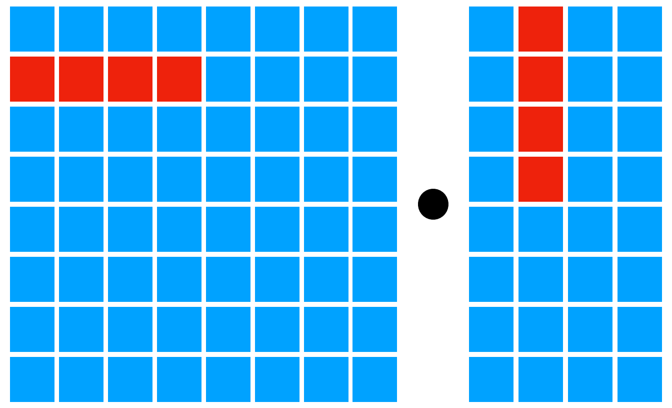
vector

Tensorization Challenge

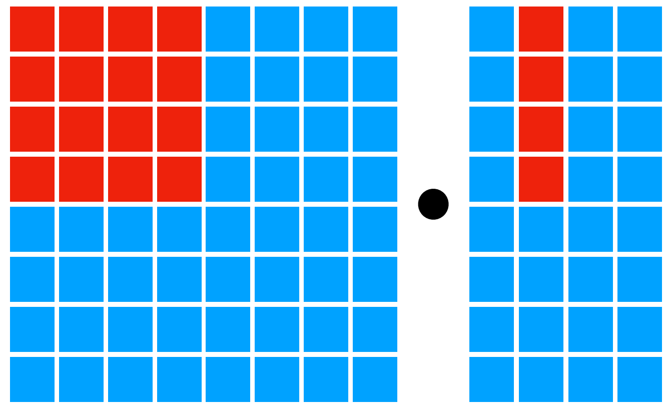
Compute primitives



scalar



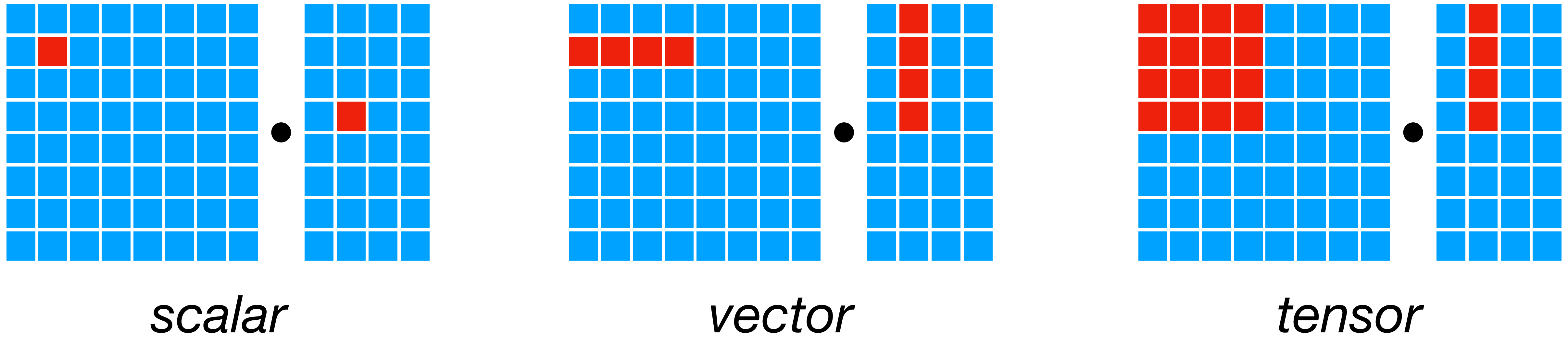
vector



tensor

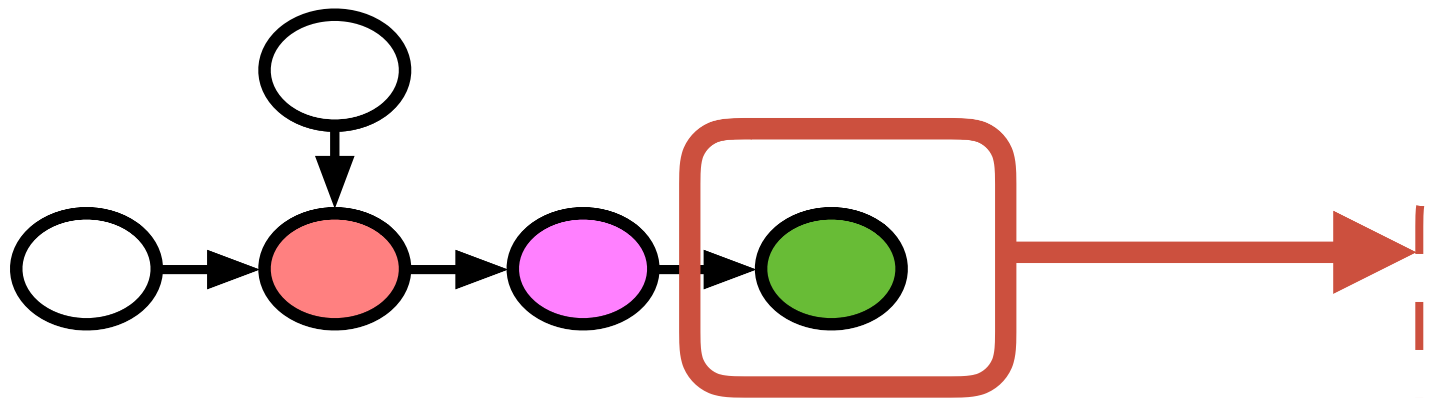
Tensorization Challenge

Compute primitives



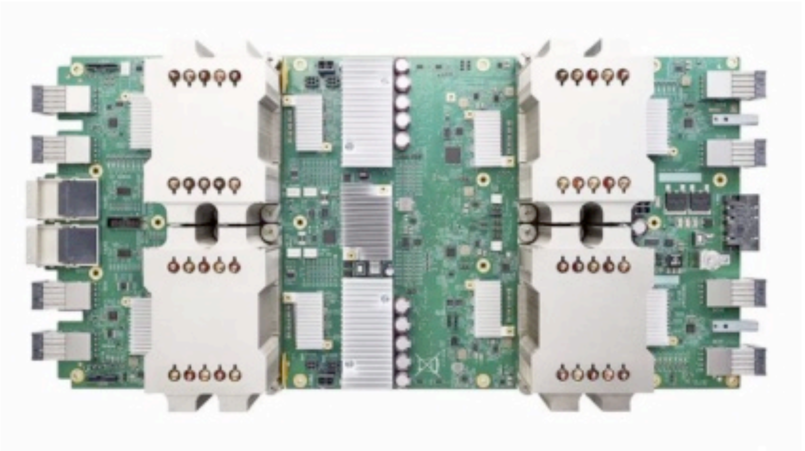
Challenge: Build systems to support emerging tensor instructions

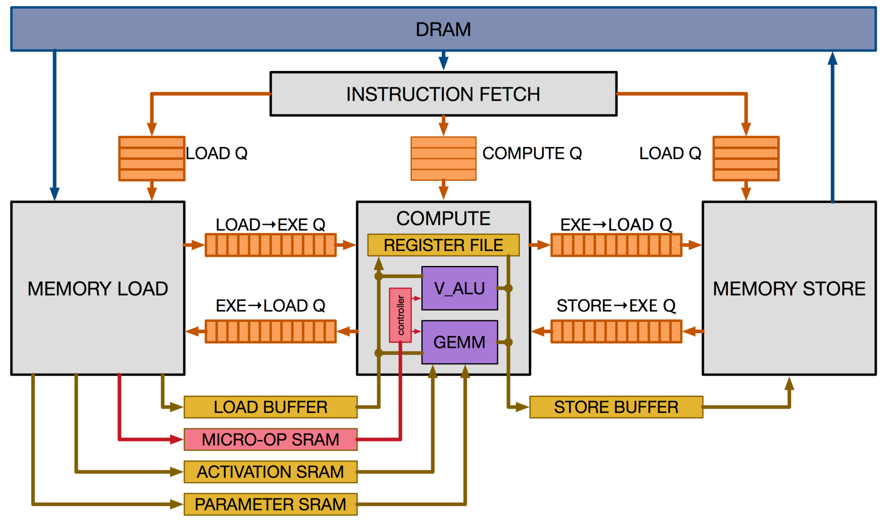
Tensorization Challenge



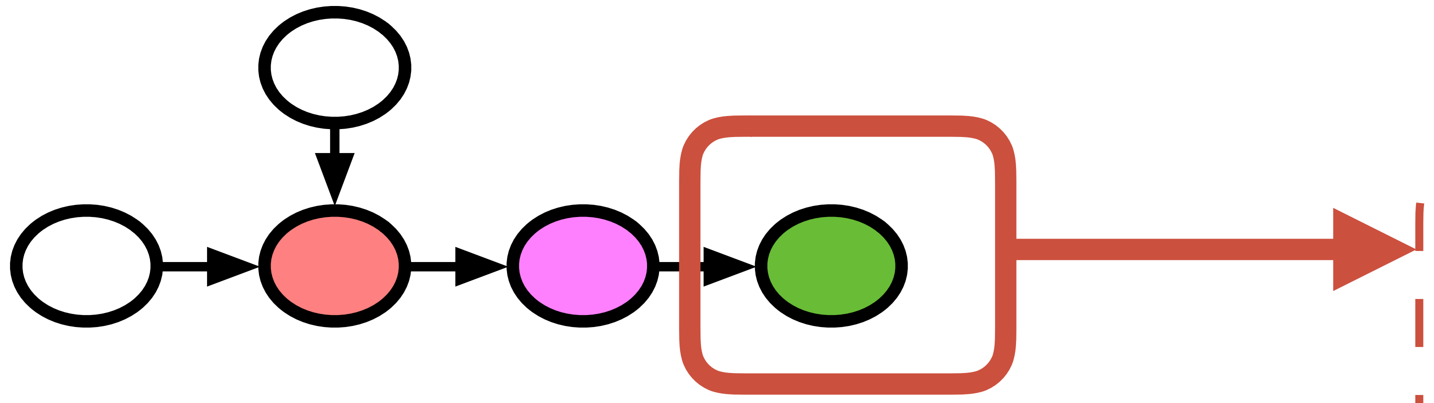
Computation Specification (Tensor Expression)

```
C = tvm.compute((m, n),
    lambda y, x: tvm.sum(A[k, y] * B[k, x], axis=k))
```



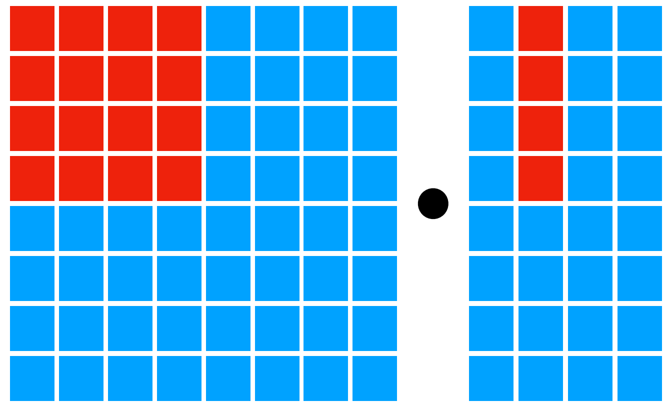


Tensorization Challenge



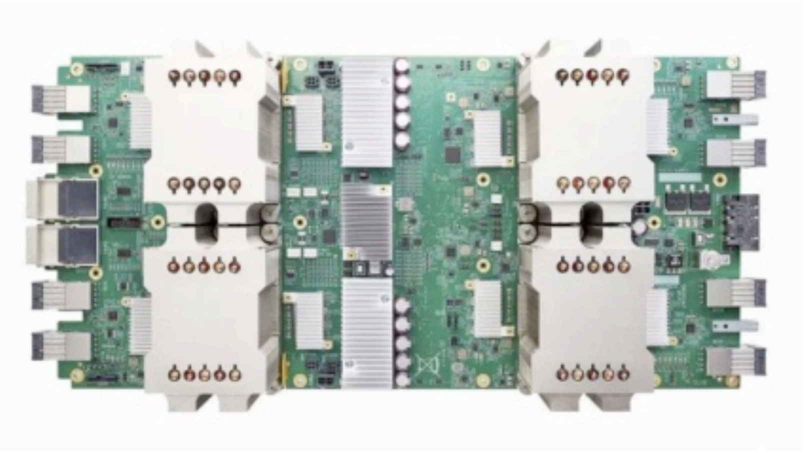
Computation Specification (Tensor Expression)

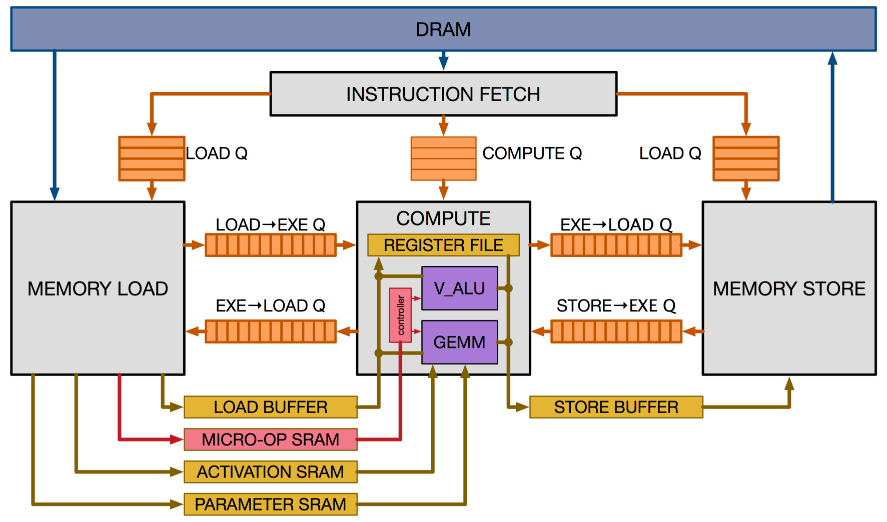
```
C = tvm.compute((m, n),
    lambda y, x: tvm.sum(A[k, y] * B[k, x], axis=k))
```



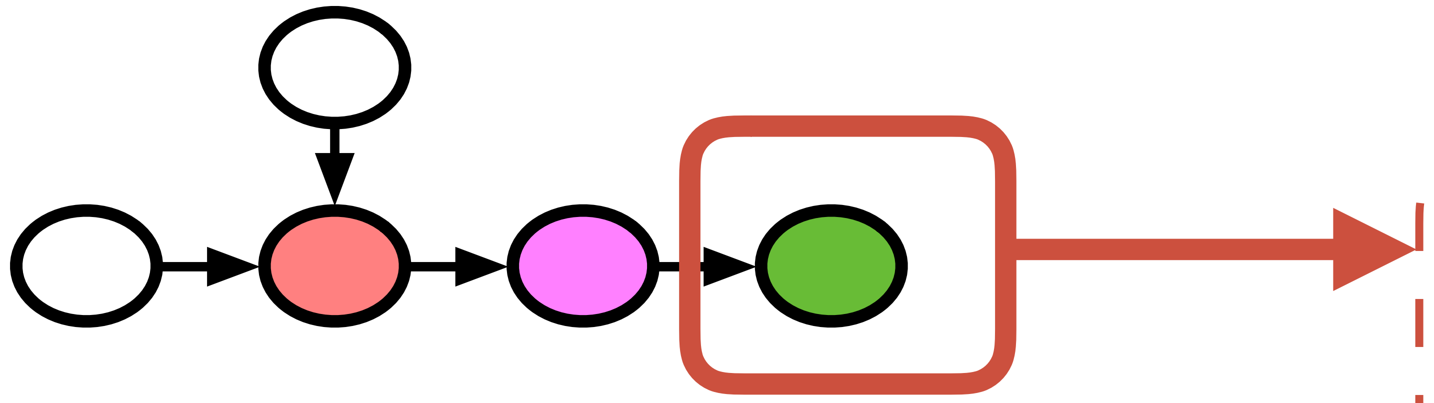
```
A = tvm.placeholder((8, 8))
B = tvm.placeholder((8,))
k = tvm.reduce_axis((0, 8))
C = tvm.compute((8, 8),
    lambda y, x: tvm.sum(A[k, y] * B[k], axis=k))
```

HW Interface Specification by Tensor Expression



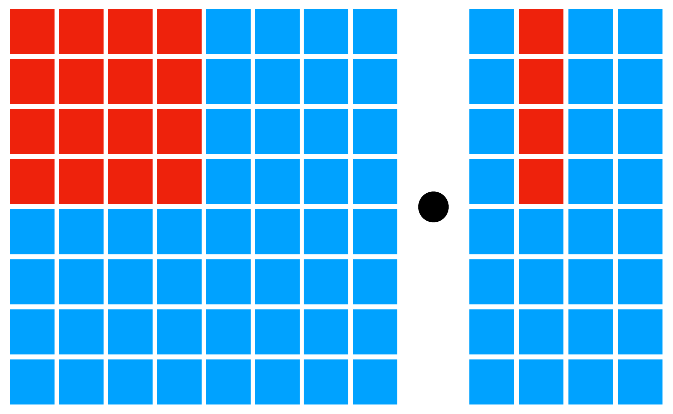


Tensorization Challenge



Computation Specification (Tensor Expression)

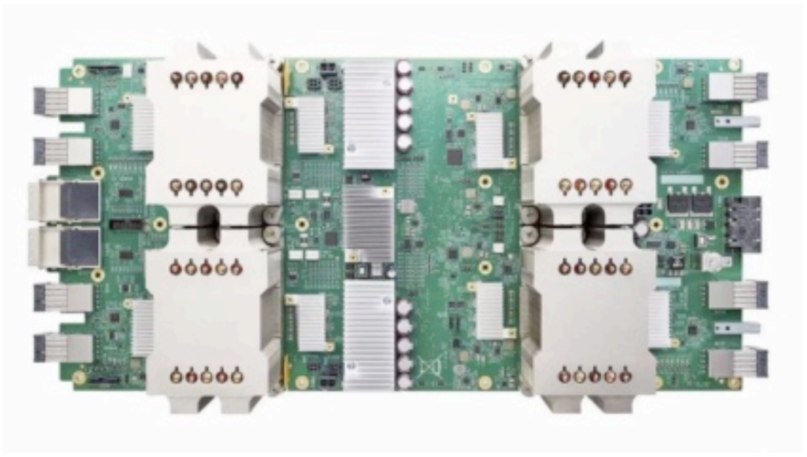
```
C = tvm.compute((m, n),
    lambda y, x: tvm.sum(A[k, y] * B[k, x], axis=k))
```

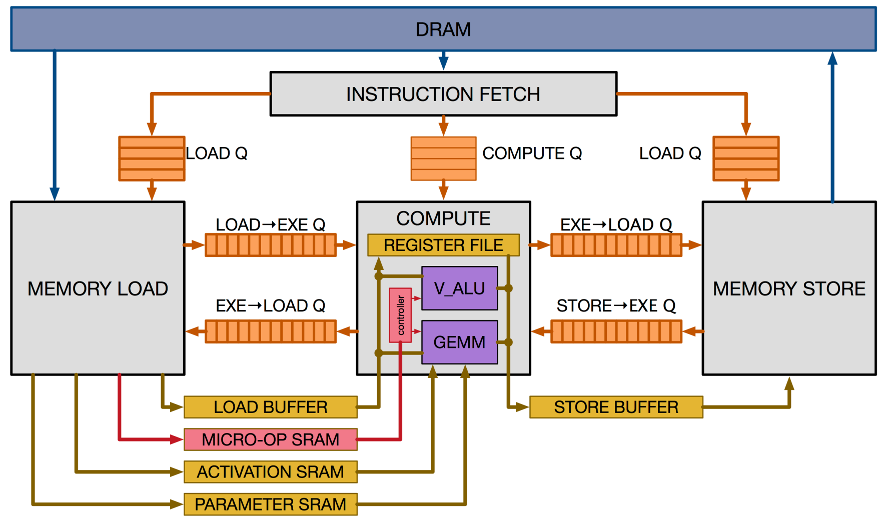


```
A = tvm.placeholder((8, 8))
B = tvm.placeholder((8,))
k = tvm.reduce_axis((0, 8))
C = tvm.compute((8, 8),
    lambda y, x: tvm.sum(A[k, y] * B[k], axis=k))
```

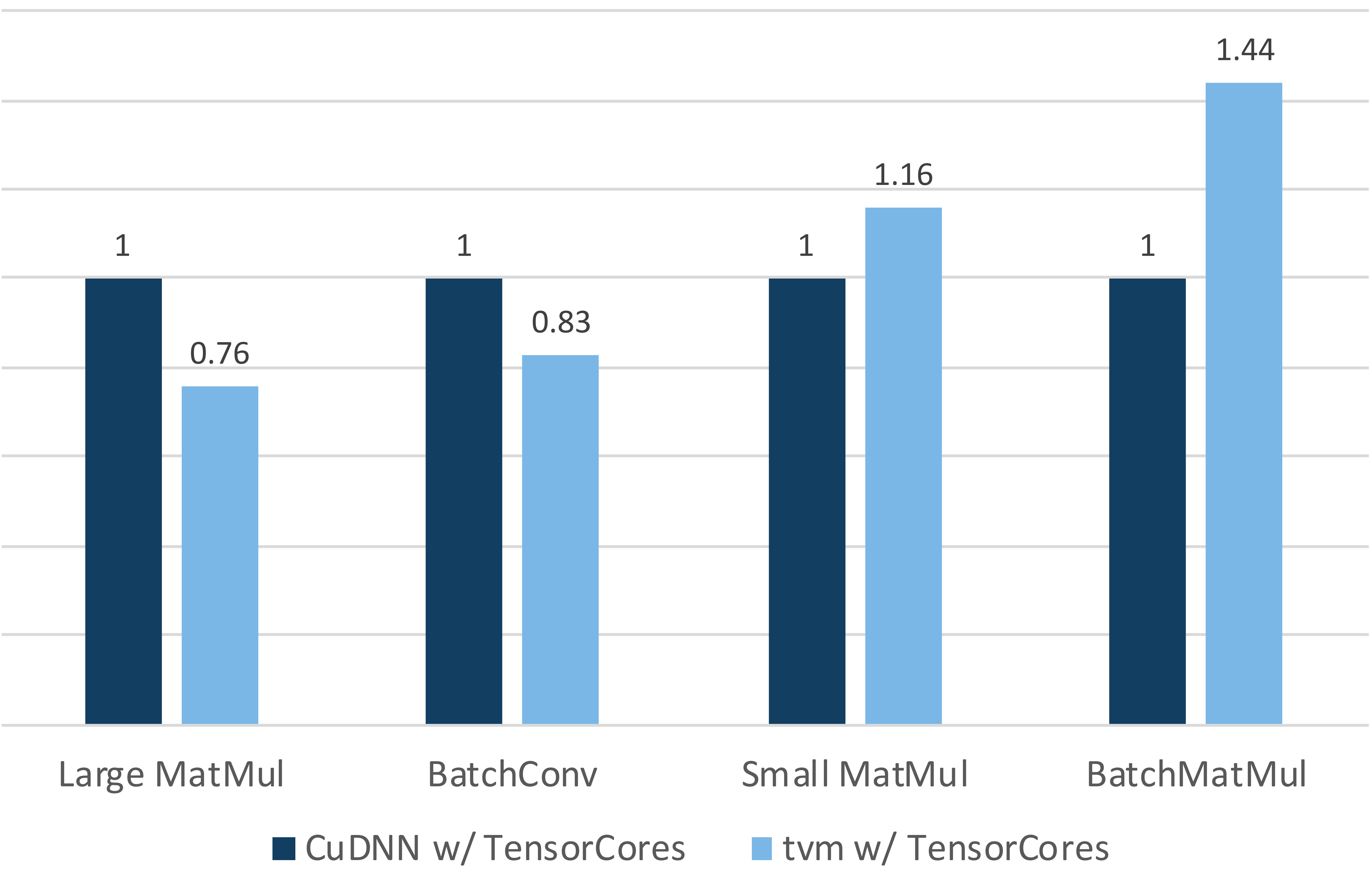
Tensorization

HW Interface Specification by Tensor Expression



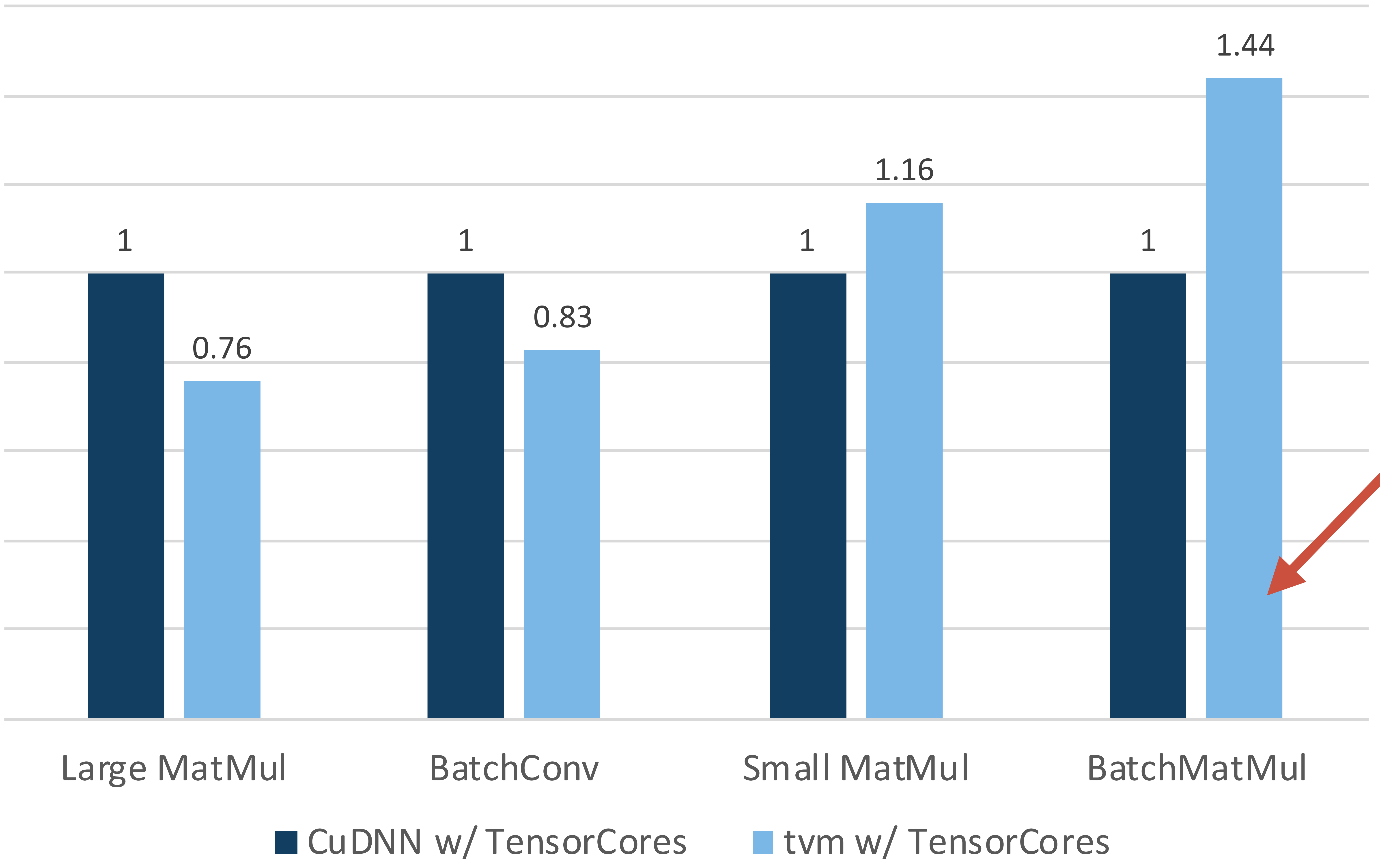


TVM for TensorCore



Credit: Siyuan Feng

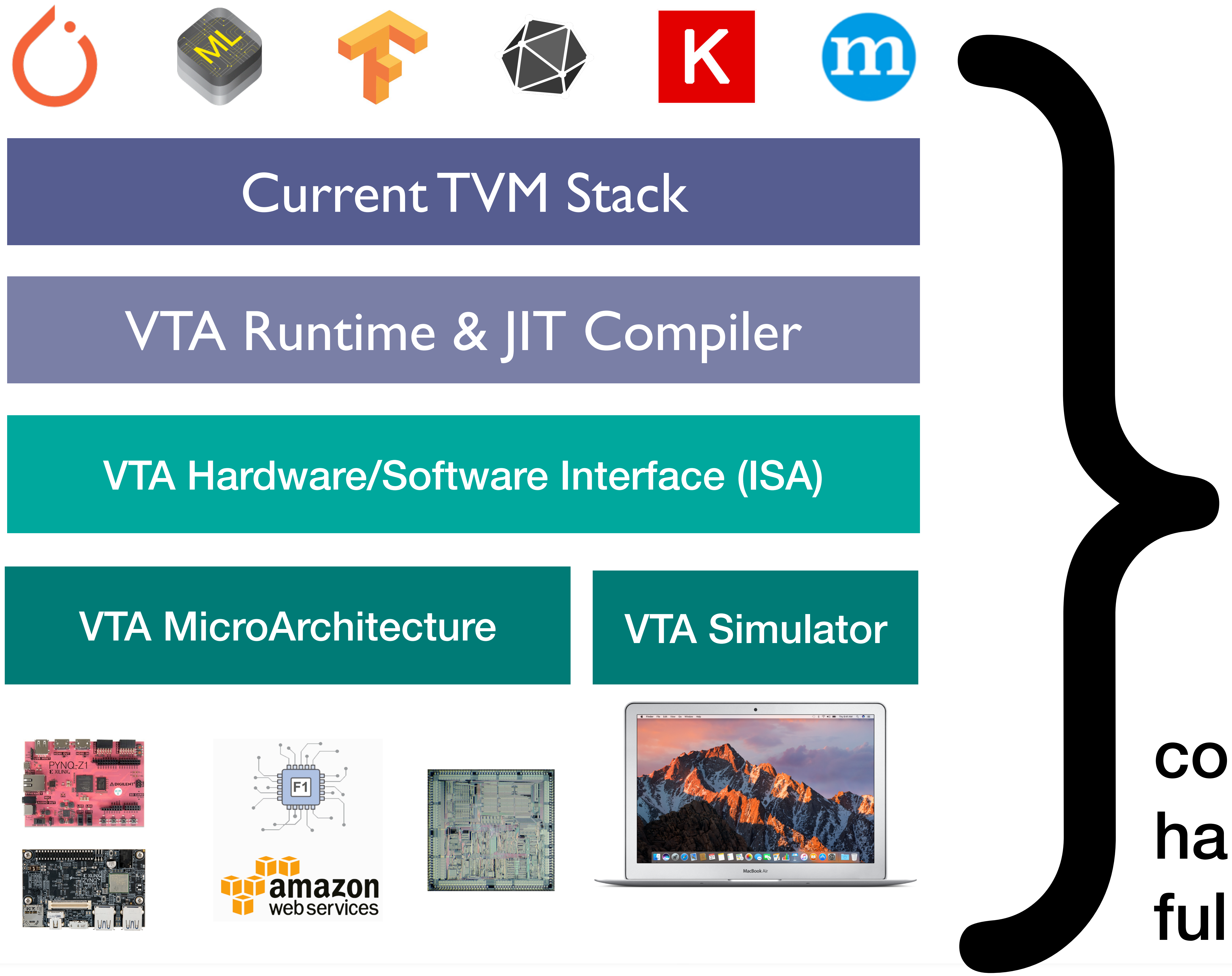
TVM for TensorCore



1.4x better on emerging workloads Transformer related workloads

Credit: Siyuan Feng

VTA: Open & Flexible Deep Learning Accelerator



Current TVM Stack

VTA Runtime & JIT Compiler

VTA Hardware/Software Interface (ISA)

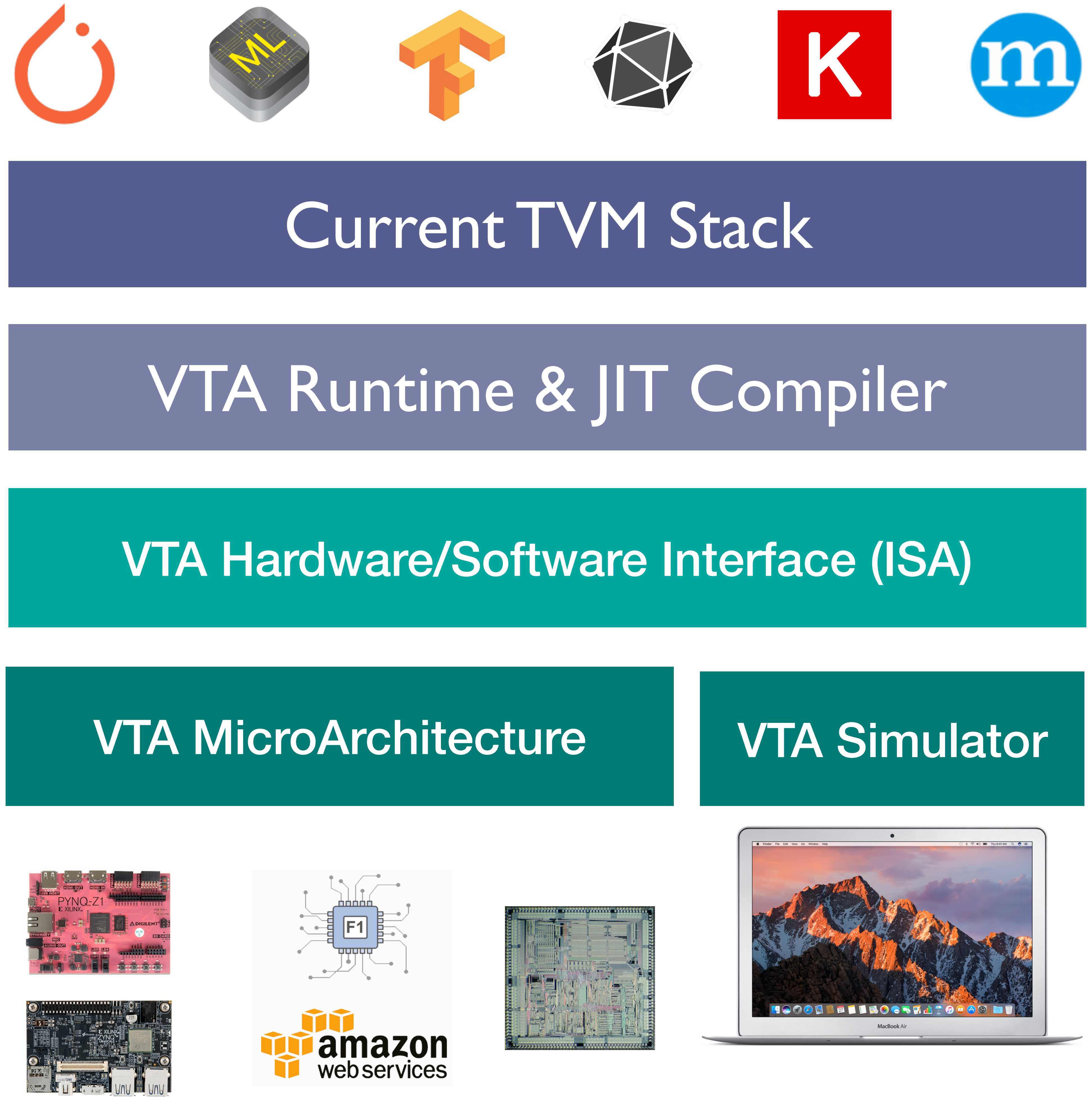
VTA MicroArchitecture

VTA Simulator

compiler, driver,
hardware design
full stack open source

HW-SW Blueprint for Flexible Deep Learning Acceleration. **Moreau et al. IEEE Micro 2019.**

VTA: Open & Flexible Deep Learning Accelerator



Current TVM Stack

VTA Runtime & JIT Compiler

VTA Hardware/Software Interface (ISA)

VTA MicroArchitecture

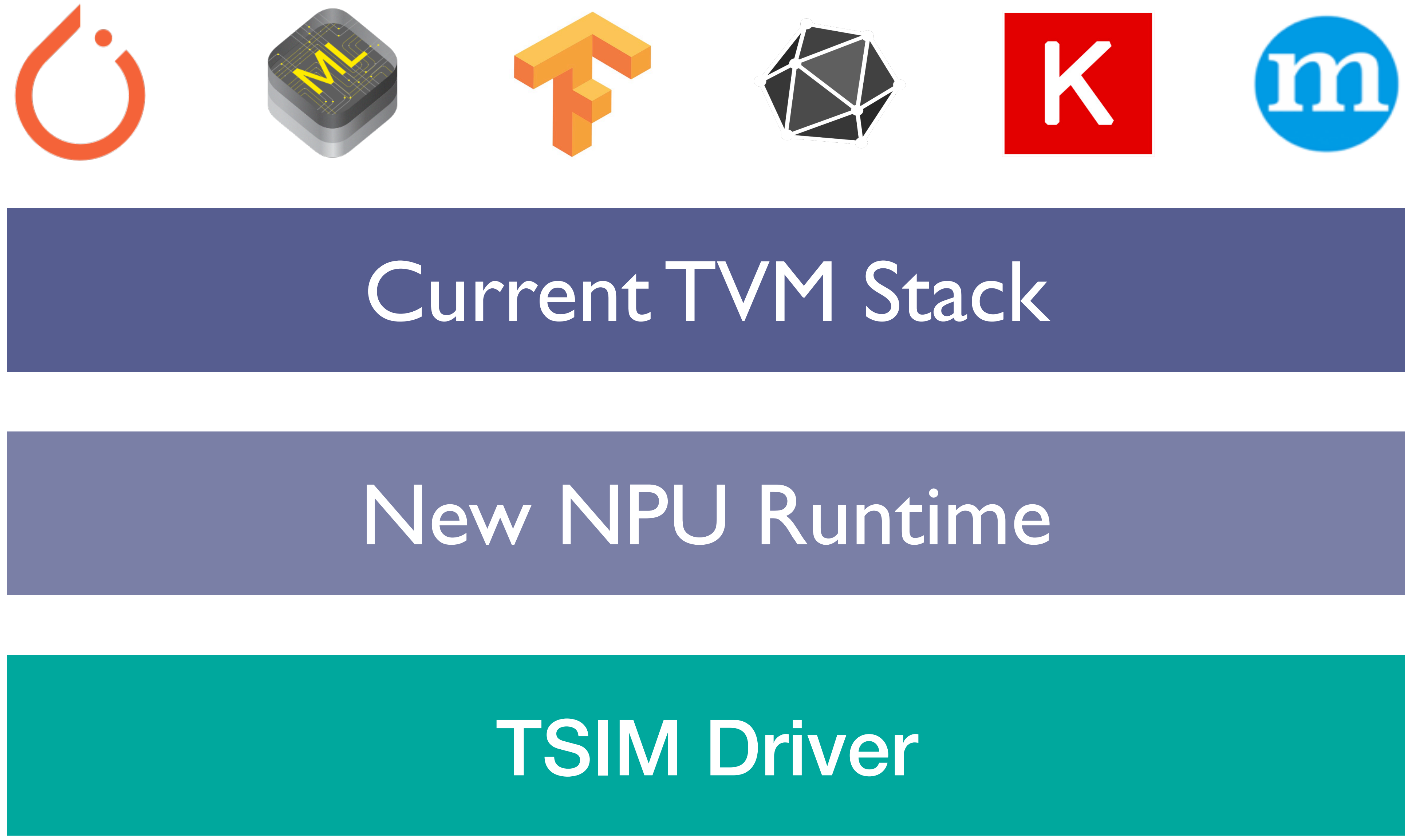
VTA Simulator

- Runtime JIT compile accelerator micro code
- Support heterogenous devices, 10x better than CPU on the same board.
- Move hardware complexity to software
- VTA 2.0 release - Chisel

compiler, driver,
hardware design
full stack open source

HW-SW Blueprint for Flexible Deep Learning Acceleration. **Moreau et al. IEEE Micro 2019.**

TSIM: Support for Future Hardware



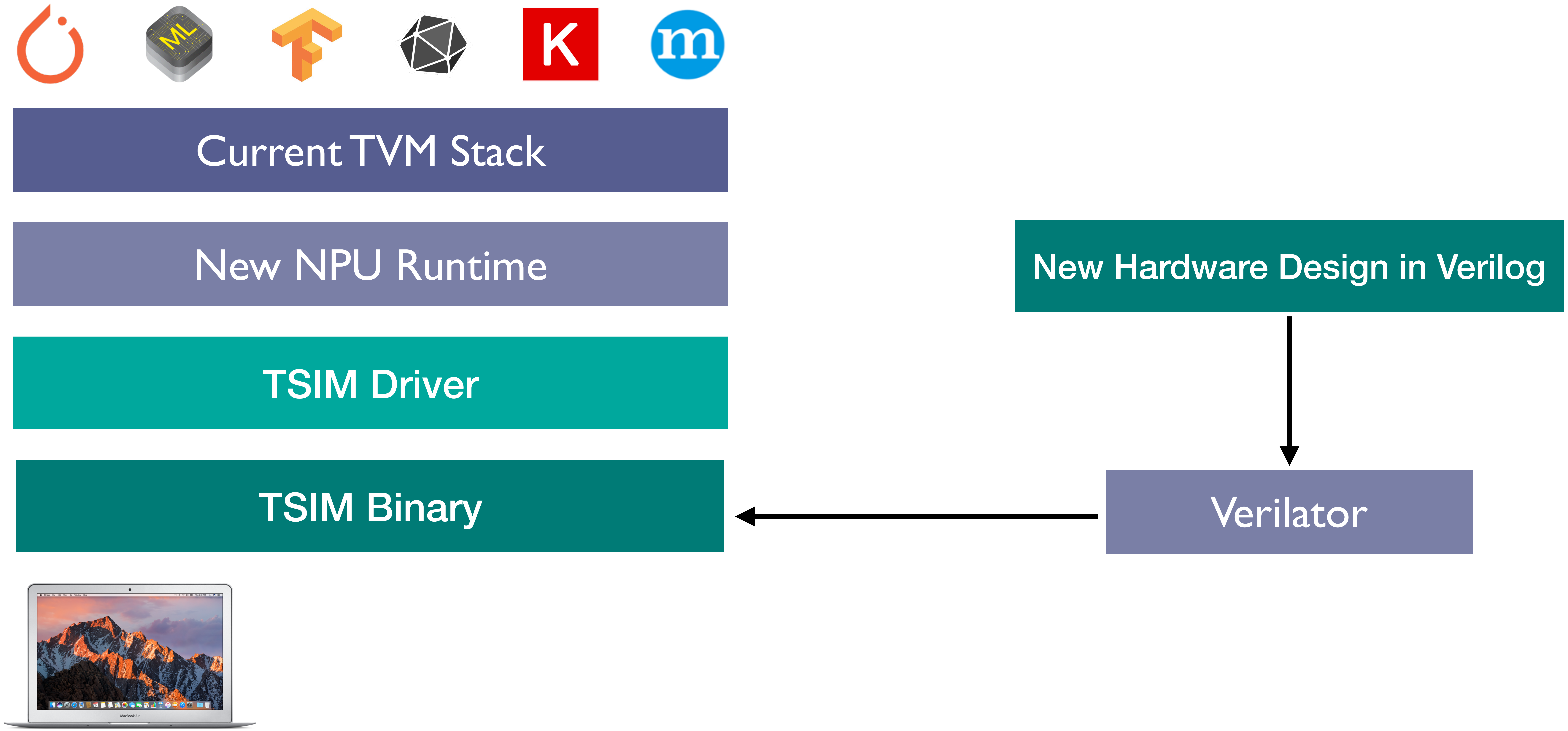
Current TVM Stack

New NPU Runtime

TSIM Driver

Credit: Luis Vega, Thierry Moureau

TSIM: Support for Future Hardware



Current TVM Stack

New NPU Runtime

TSIM Driver

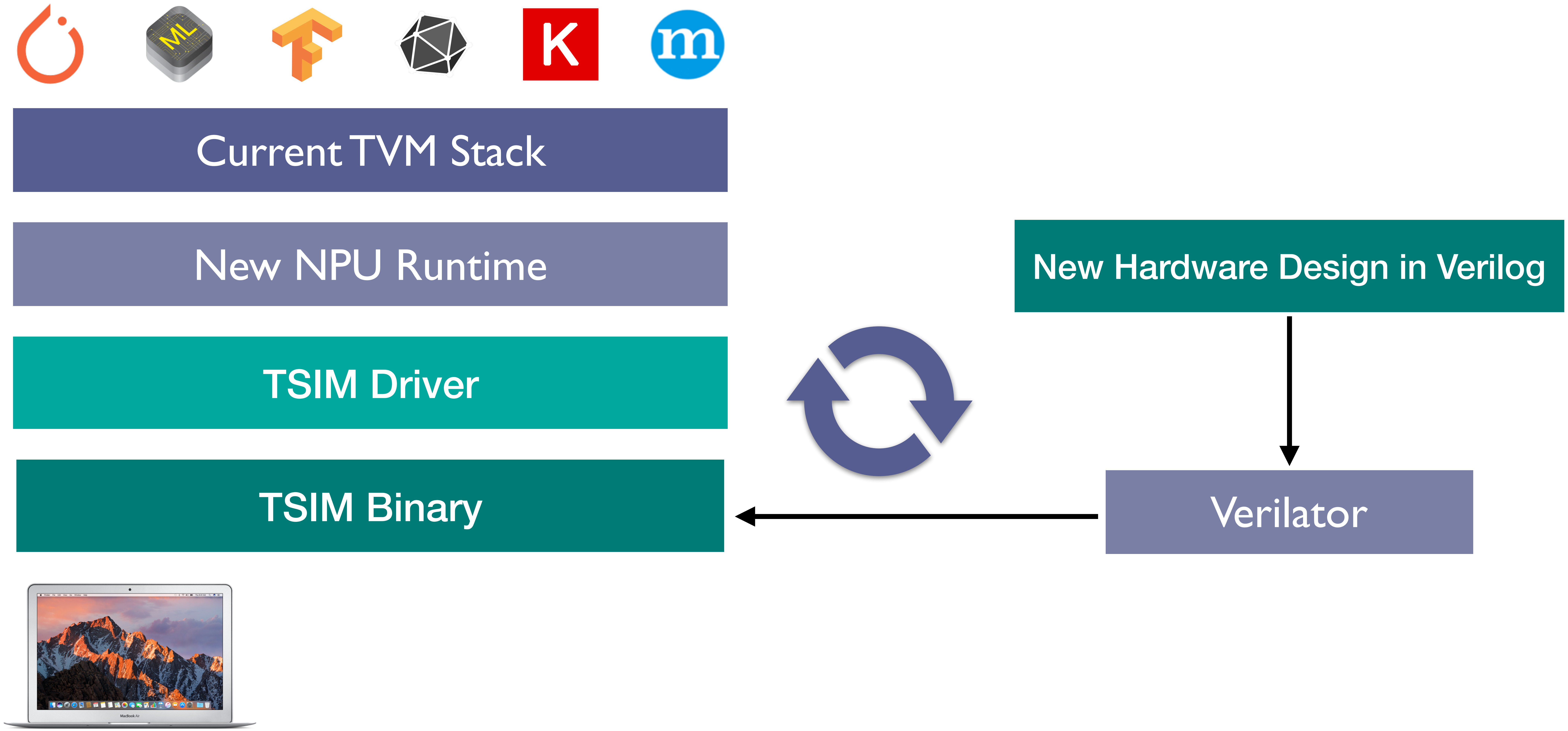
TSIM Binary

New Hardware Design in Verilog

Verilator

Credit: Luis Vega, Thierry Moureau

TSIM: Support for Future Hardware



Current TVM Stack

New NPU Runtime

TSIM Driver

TSIM Binary

New Hardware Design in Verilog

Verilator

Credit: Luis Vega, Thierry Moureau

Where are we going: Selected Topics

Unified Runtime

Unified IR

Full-stack Automation

Where are we going: Selected Topics

Unified Runtime

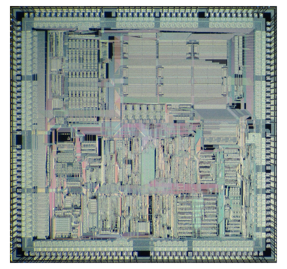
Unified IR

Full-stack Automation

Unified Runtime For Heterogeneous Devices

Device Drivers

NPU Driver



CUDA Driver

External Runtimes

TensorRT

Unified Runtime For Heterogeneous Devices

tvm::runtime::Module

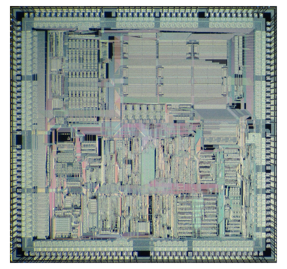
Runtime Module Interface

GetFunction(string) -> tvm::runtime::PackedFunc

SaveToBinary/LoadFromBinary

Device Drivers

NPU Driver



CUDA Driver

External Runtimes

TensorRT

Unified Runtime For Heterogeneous Devices

tvm::runtime::Module

Runtime Module Interface

GetFunction(string) -> tvm::runtime::PackedFunc

SaveToBinary/LoadFromBinary

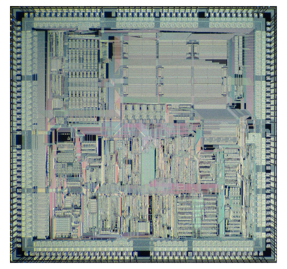
NPUModule

CUDAModule

TFModule

Device Drivers

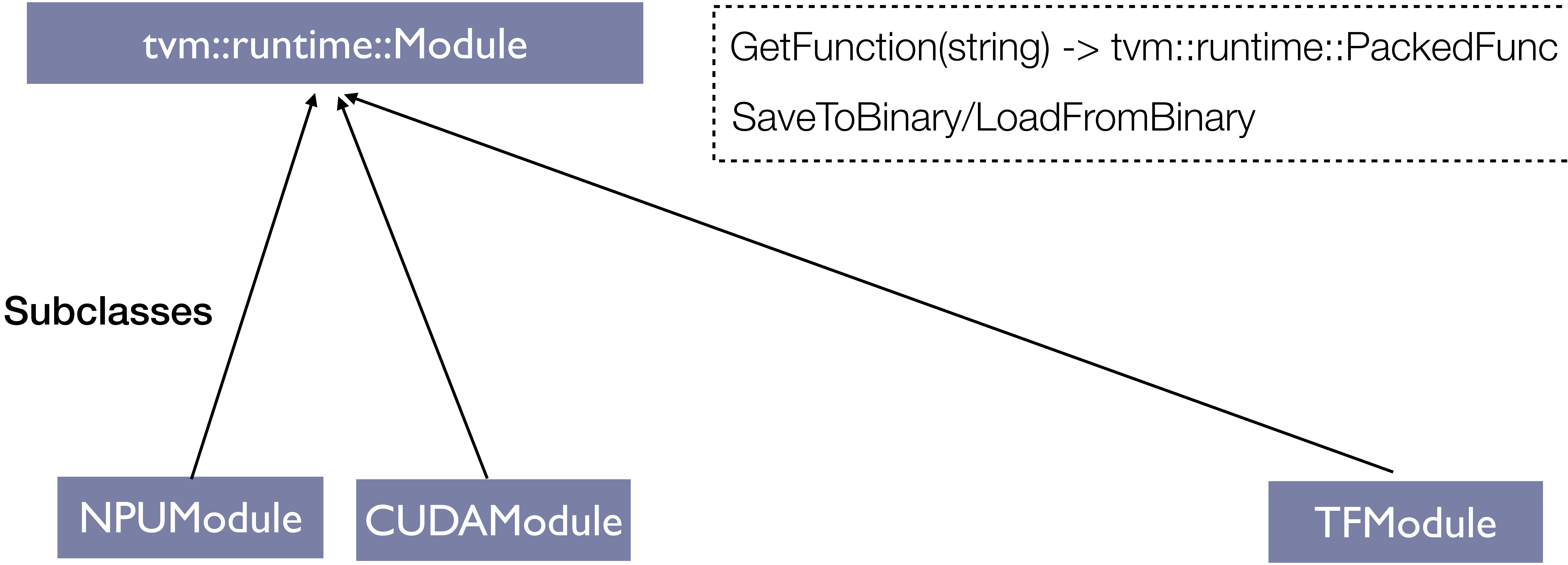
NPU Driver



CUDA Driver

External Runtimes

Unified Runtime For Heterogeneous Devices



tvm::runtime::Module

Subclasses

NPUModule

CUDAModule

TFModule

Runtime Module Interface

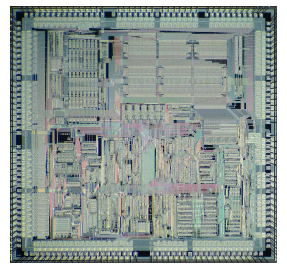
GetFunction(string) -> tvm::runtime::PackedFunc

SaveToBinary/LoadFromBinary

NPU Driver

CUDA Driver

Device Drivers



External Runtimes

TensorRT

Unified Runtime Benefit

Unified library packaging

```
mod.export_library("mylib.so")
```

Free API (Py/Java/Go)

```
lib = tvm.module.load("mylib.so")
func = lib["npufunction0"]
func(a, b)
```

Automatic RPC Support

```
remote = tvm.rpc.connect(board_url, port)
remote.upload("mylib.so")
remote_mod = remote.load_module(“mylib.so")
func = remote_mod[“npufunction0"]
func(remote_a, remote_b)
```

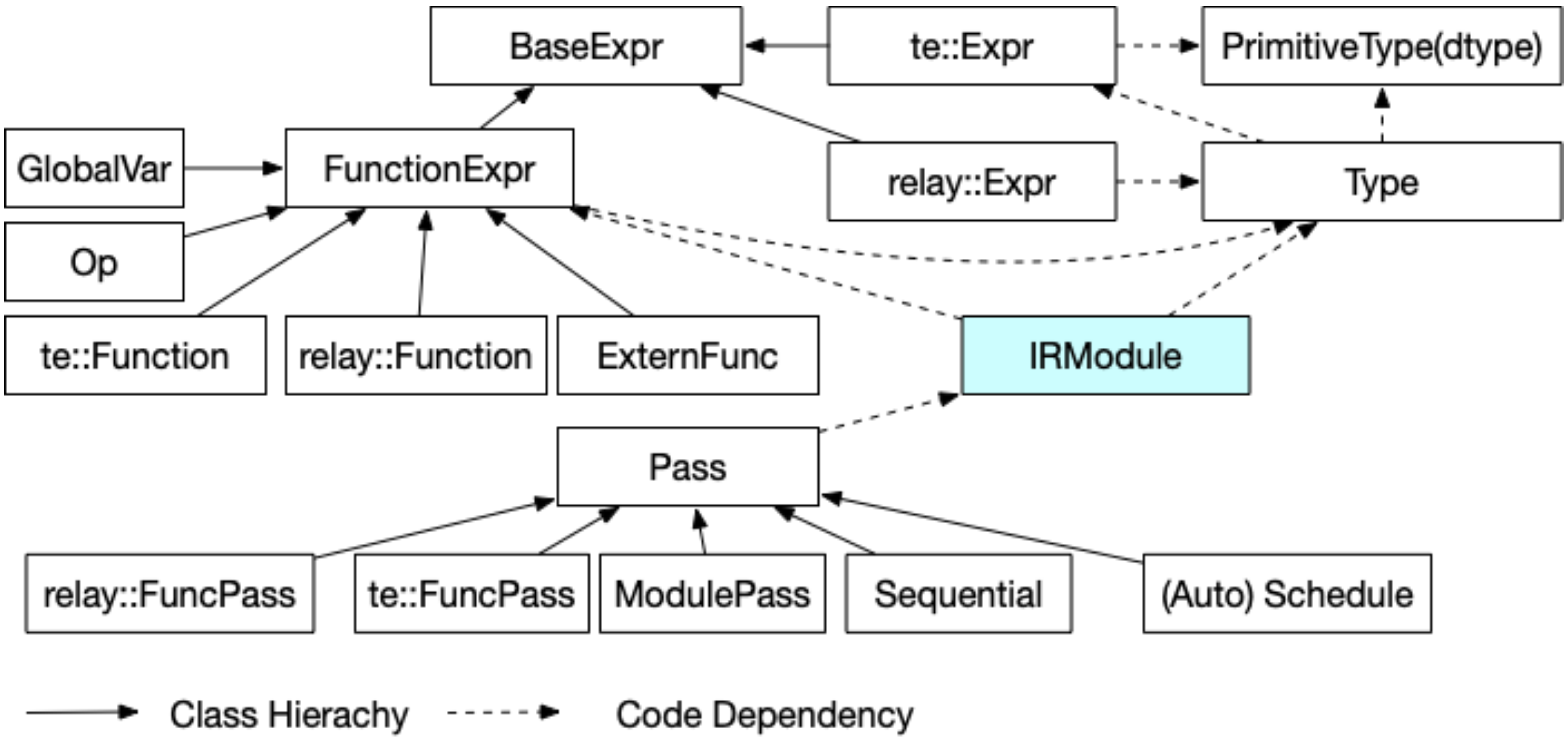
Where are we going: Selected Topics

Unified Runtime

Unified IR

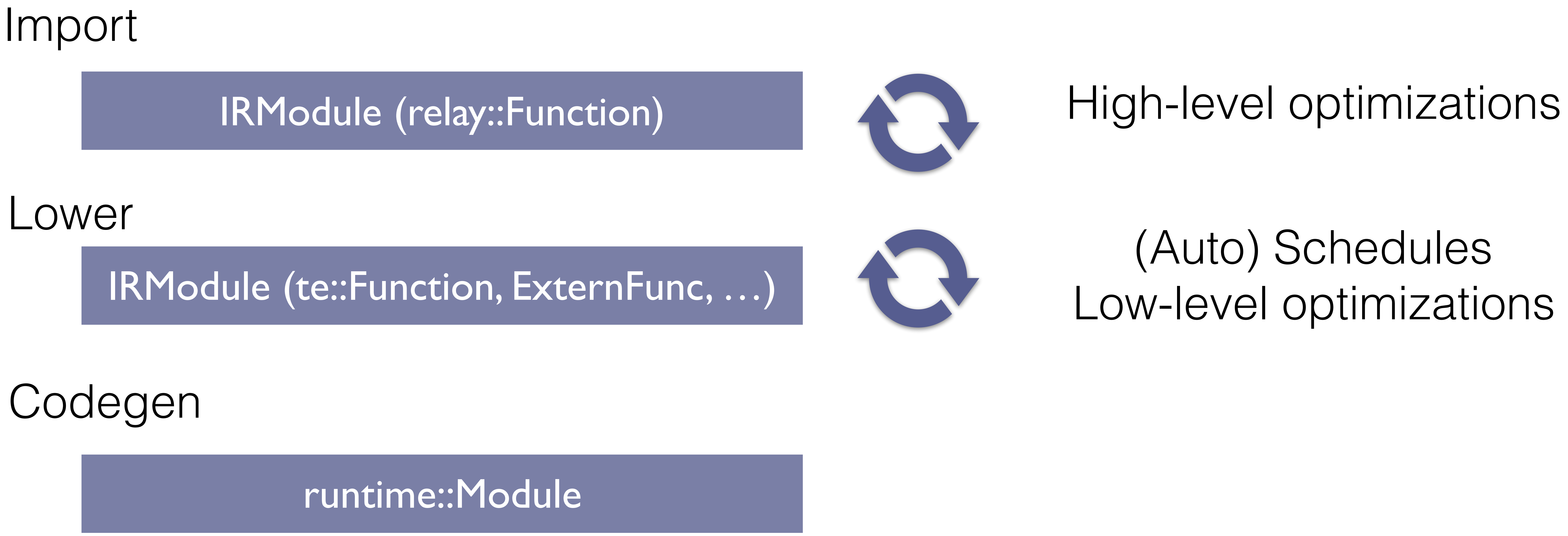
Full-stack Automation

Overview of New IR Infra



Unified module/pass, type system, with function variants support

Compilation Flow under the New Infra



Import

IRModule (relay::Function)

High-level optimizations

Lower

IRModule (te::Function, ExternFunc, …)

(Auto) Schedules
Low-level optimizations

Codegen

runtime::Module

Mixed Function Variants in the Same Module

```
def @relay_add_one(%x : Tensor((10,), f32)) {
    call_destination_passing @te_add_one(%x,  out=%b)
}

def @te_add_one(%a: NDArray, %b: NDArray) {
    var %n
    %A = decl_buffer(shape=[%n], src=%a)
    %B = decl_buffer(shape=[%n], src=%b)
    for %i = 0 to 10 [data_par] {
        %B[%i] = %A[%i] + 1.0
    }
}
```

First-class Python Support

```
@tvm.hybrid
def te_add_one(a, b):
   n = var(“n”)
   A = bind_buffer(shape=[n], a)
   B = bind_buffer(shape=[n], b)
   for i in iter_range(n, iter_type=”data_par”):
      A[i] = B[i] + 1


mod = tvm.IRModule([te_add_one])
print(mod[”te_add_one”].args)
```

First-class Python Support

```
@tvm.hybrid
def te_add_one(a, b):
    n = var(“n”)
    A = bind_buffer(shape=[n], a)
    B = bind_buffer(shape=[n], b)
    for i in iter_range(n, iter_type=”data_par”):
        A[i] = B[i] + 1


mod = tvm.IRModule([te_add_one])
print(mod[”te_add_one”].args)
```

Use hybrid script as an alternative text format

First-class Python Support

```
@tvm.hybrid
def te_add_one(a, b):
    n = var(“n”)
    A = bind_buffer(shape=[n], a)
    B = bind_buffer(shape=[n], b)
    for i in iter_range(n, iter_type=”data_par”):
        A[i] = B[i] + 1


mod = tvm.IRModule([te_add_one])
print(mod[”te_add_one”].args)
```

Use hybrid script as an alternative text format

Directly write pass, manipulate IR structures

First-class Python Support

```
@tvm.hybrid
def te_add_one(a, b):
   n = var(“n”)
   A = bind_buffer(shape=[n], a)
   B = bind_buffer(shape=[n], b)
   for i in iter_range(n, iter_type=”data_par”):
      A[i] = B[i] + 1
```

Use hybrid script as an alternative text format

```
mod = tvm.IRModule([te_add_one])
print(mod[”te_add_one”].args)
```

Directly write pass, manipulate IR structures

Accelerate innovation,
e.g. use (GA/RL/BayesOpt/your favorite ML method) for AutoSchedule

Easy shift to C++ when product ready

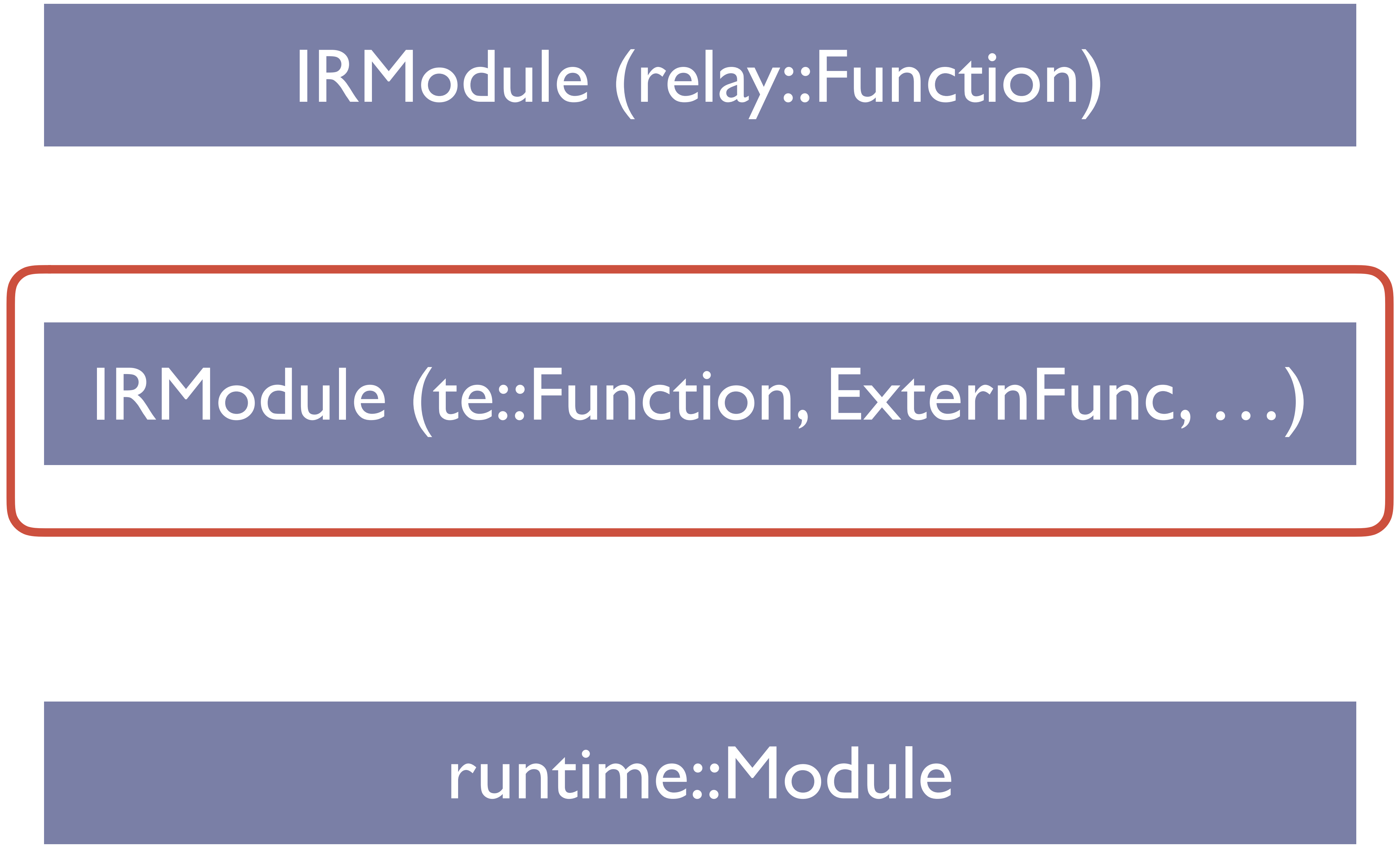
Rethink Low-level Tensor IR

IRModule (relay::Function)

IRModule (te::Function, ExternFunc, …)

runtime::Module

Rethink Low-level Tensor IR

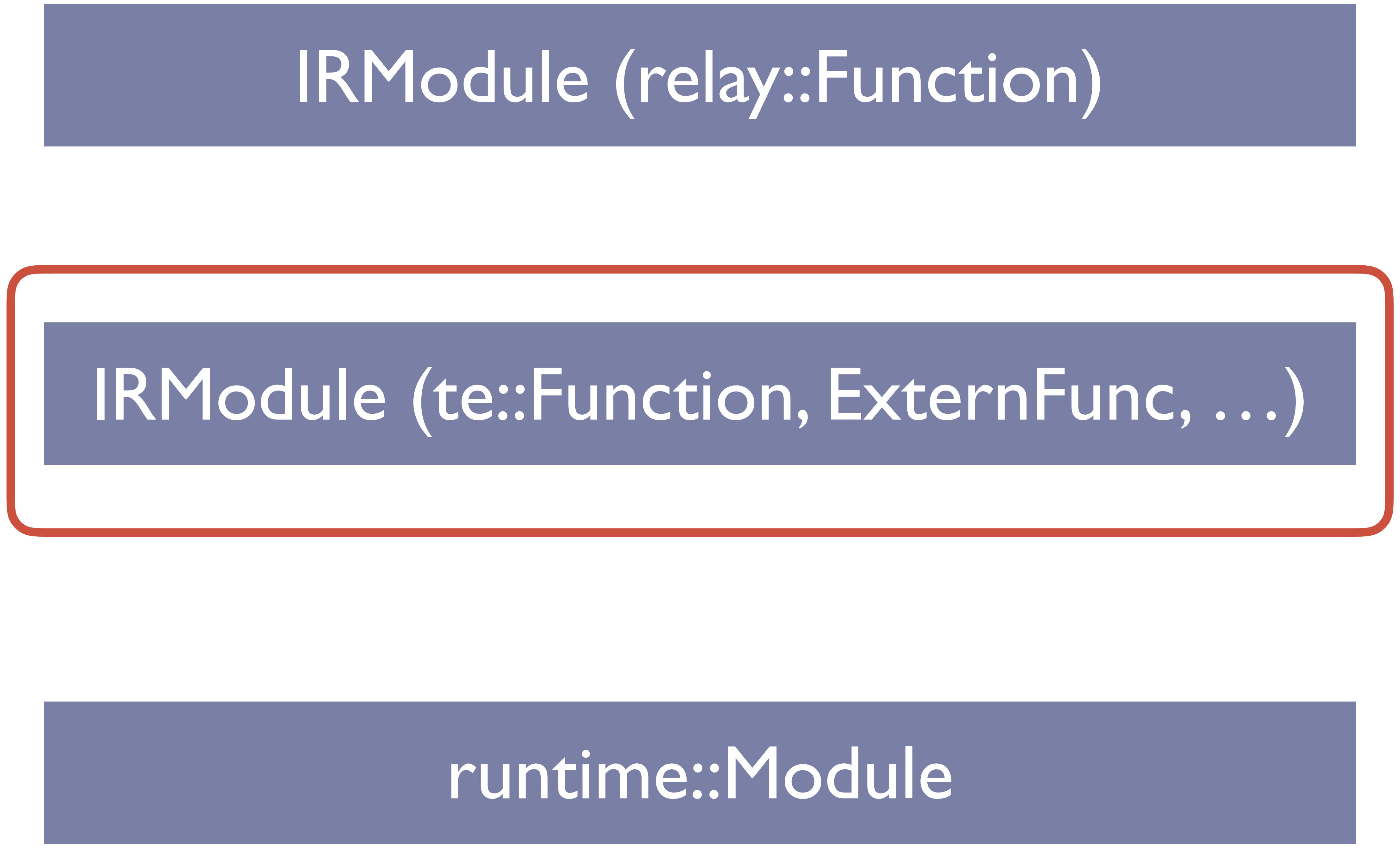


IRModule (relay::Function)

IRModule (te::Function, ExternFunc, …)

runtime::Module

Rethink Low-level Tensor IR

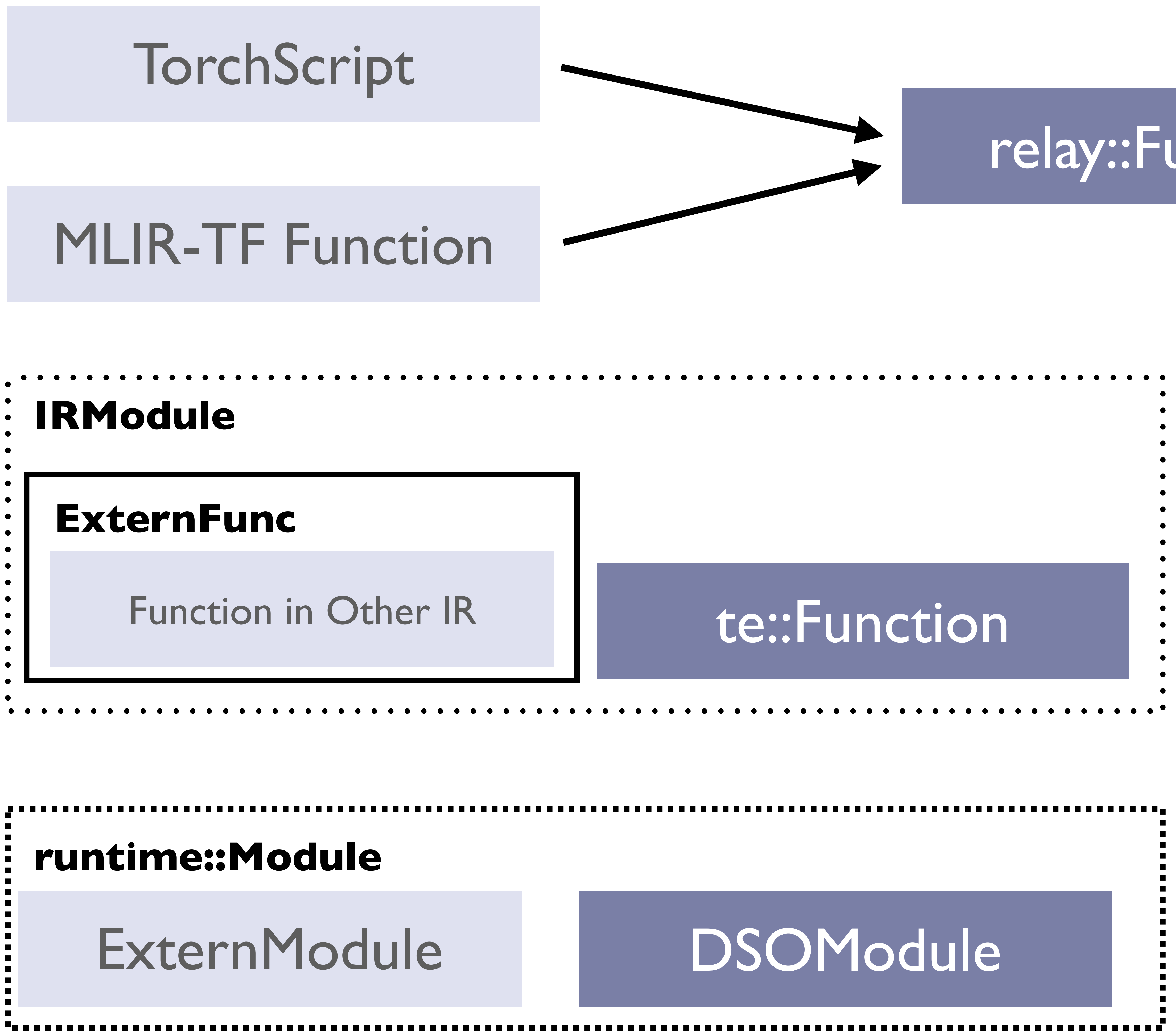

IRModule (relay::Function)
IRModule (te::Function, ExternFunc, …)
runtime::Module

Function as unit of transformation

Schedule transformation as pass

Better tensorization support

Interpolate with Other ML Compiler Infra



TorchScript

MLIR-TF Function

relay::Function

IR Translation

IRModule

ExternFunc

Function in Other IR

te::Function

Custom Packaging

runtime::Module

ExternModule

DSOModule

Custom codegen

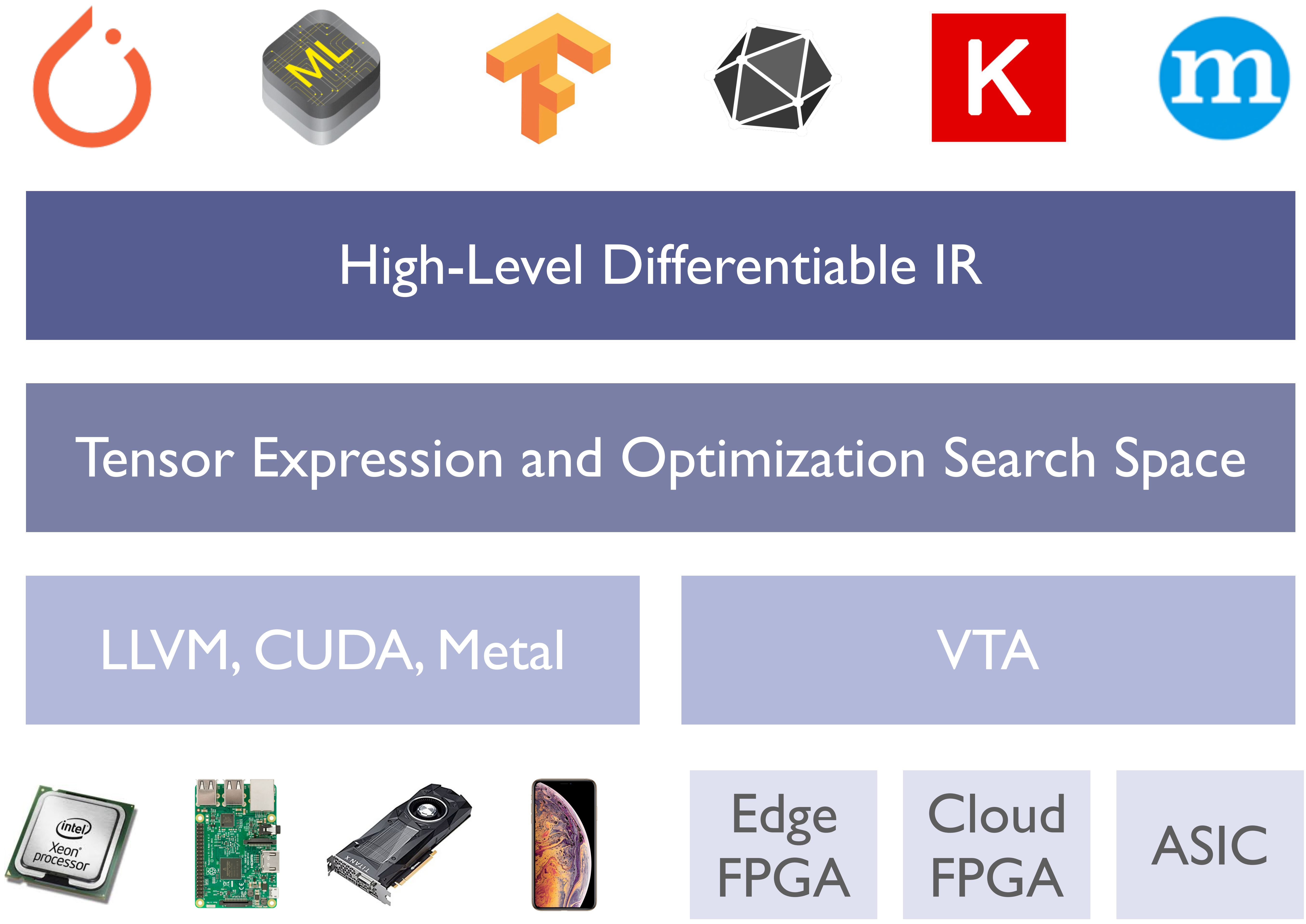
Where are we going: Selected Topics

Unified Runtime

Unified IR

Full-stack Automation

Full Stack Automation



High-Level Differentiable IR

Tensor Expression and Optimization Search Space

LLVM, CUDA, Metal

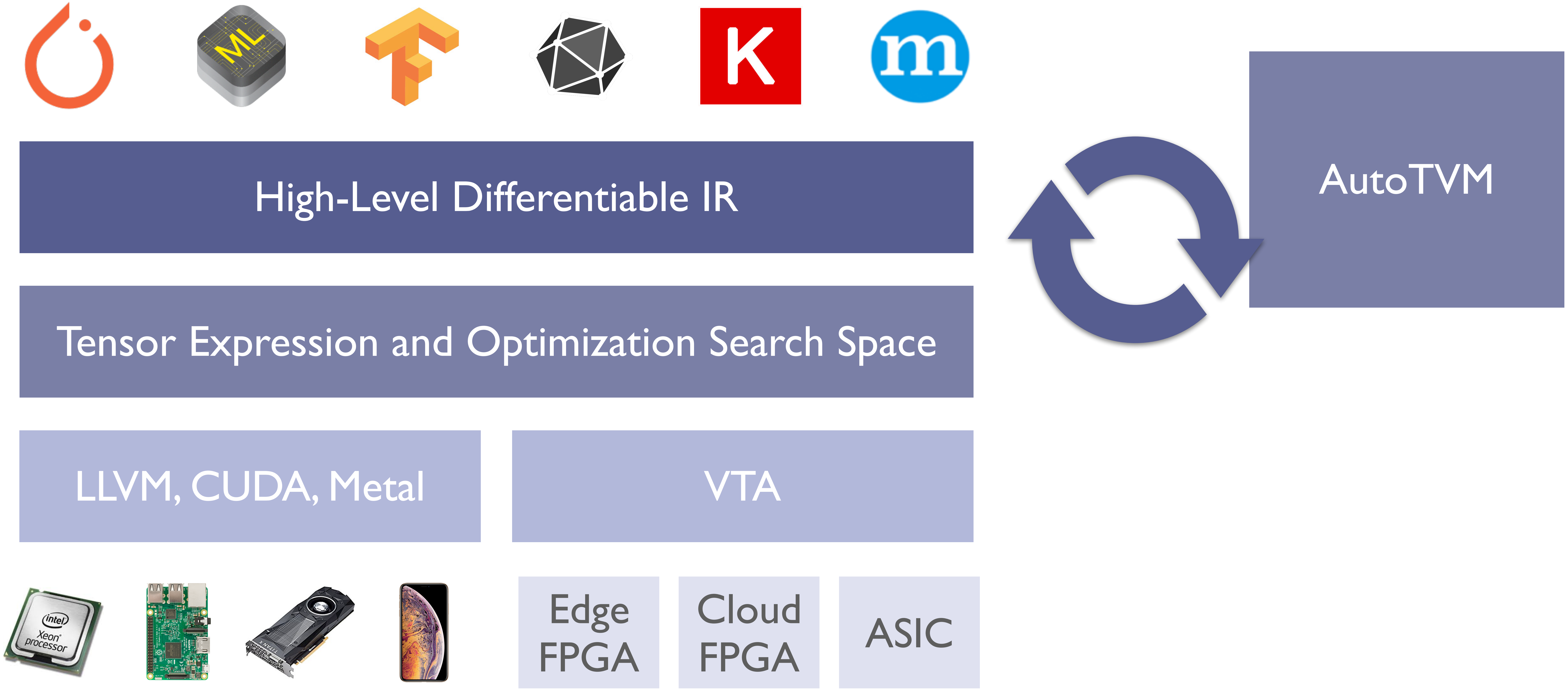
VTA

Edge FPGA

Cloud FPGA

ASIC

Full Stack Automation



High-Level Differentiable IR

Tensor Expression and Optimization Search Space

LLVM, CUDA, Metal

VTA

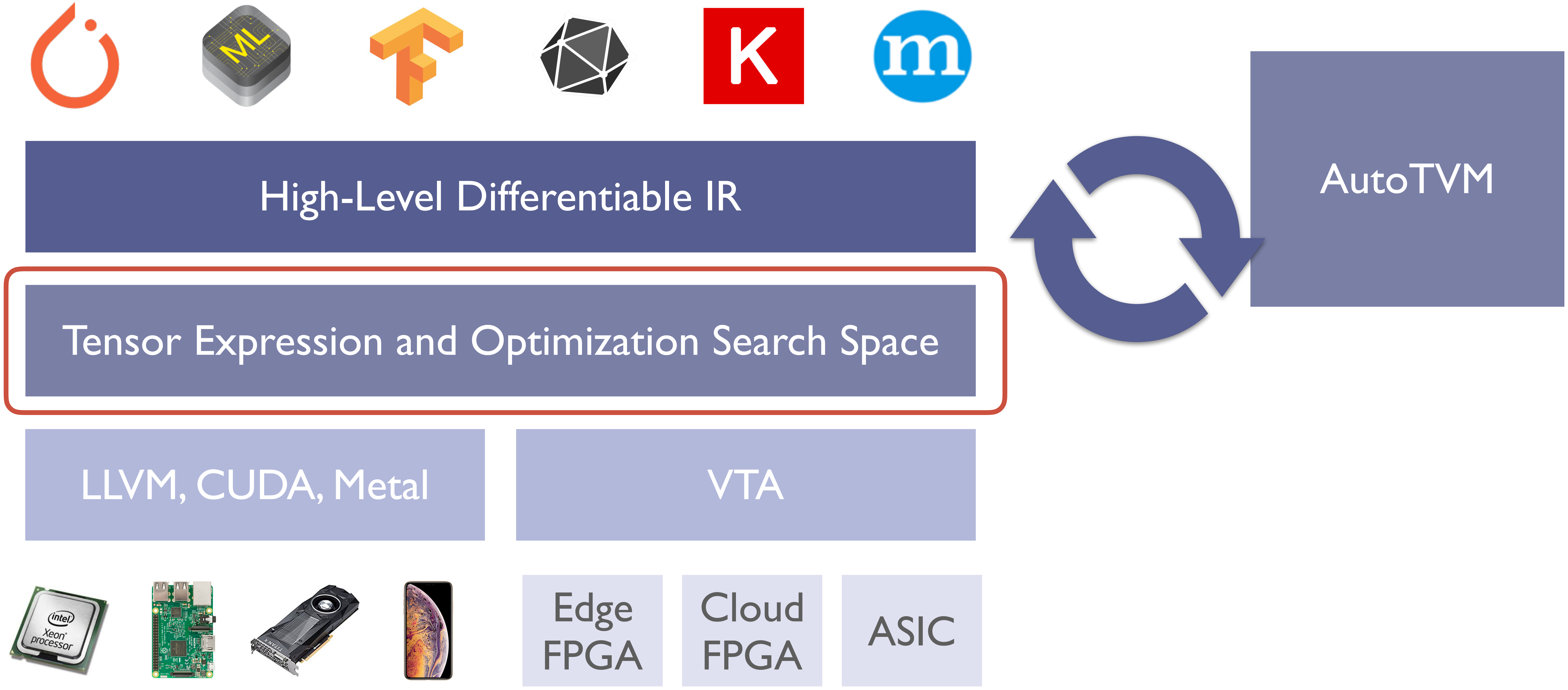
Edge FPGA

Cloud FPGA

ASIC

AutoTVM

Full Stack Automation



High-Level Differentiable IR

Tensor Expression and Optimization Search Space

LLVM, CUDA, Metal

VTA

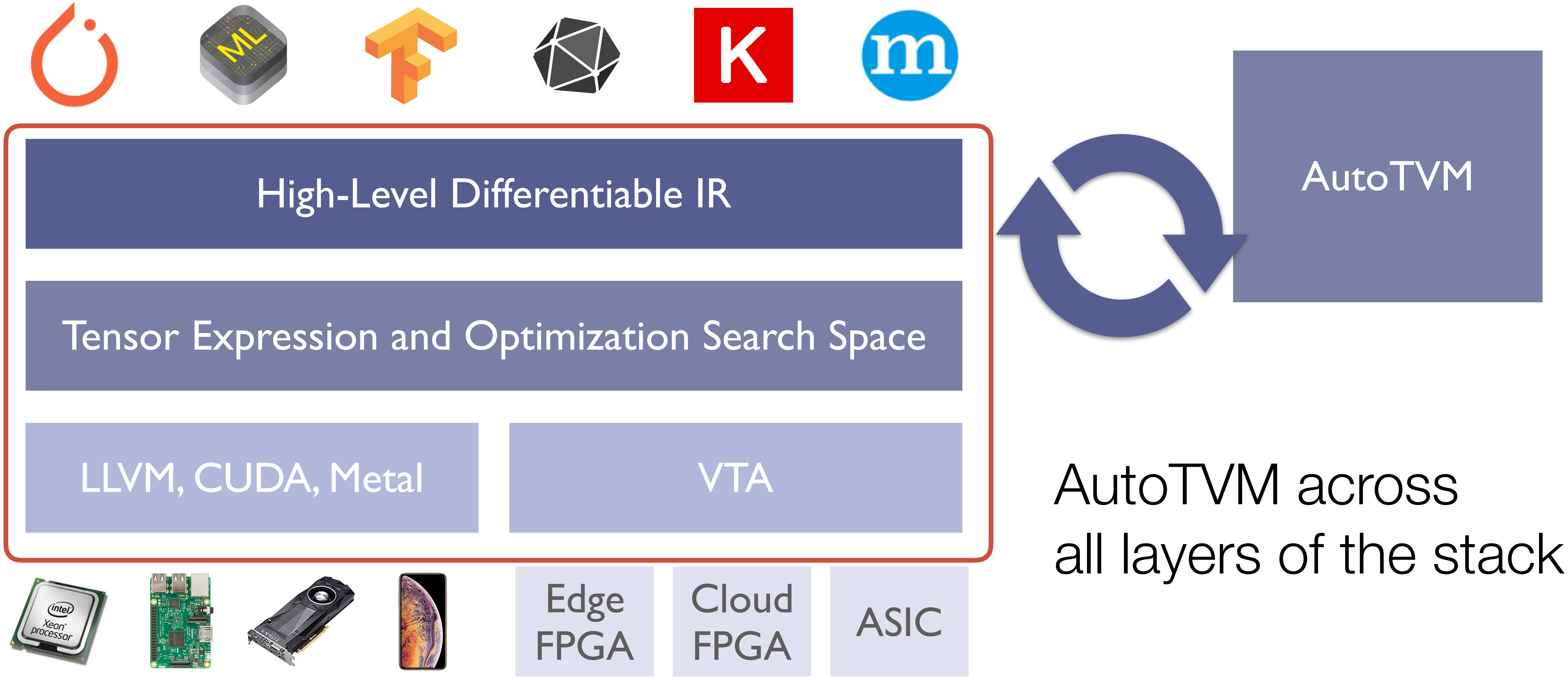
Edge FPGA

Cloud FPGA

ASIC

AutoTVM

Full Stack Automation



AutoTVM

High-Level Differentiable IR

Tensor Expression and Optimization Search Space

LLVM, CUDA, Metal

VTA

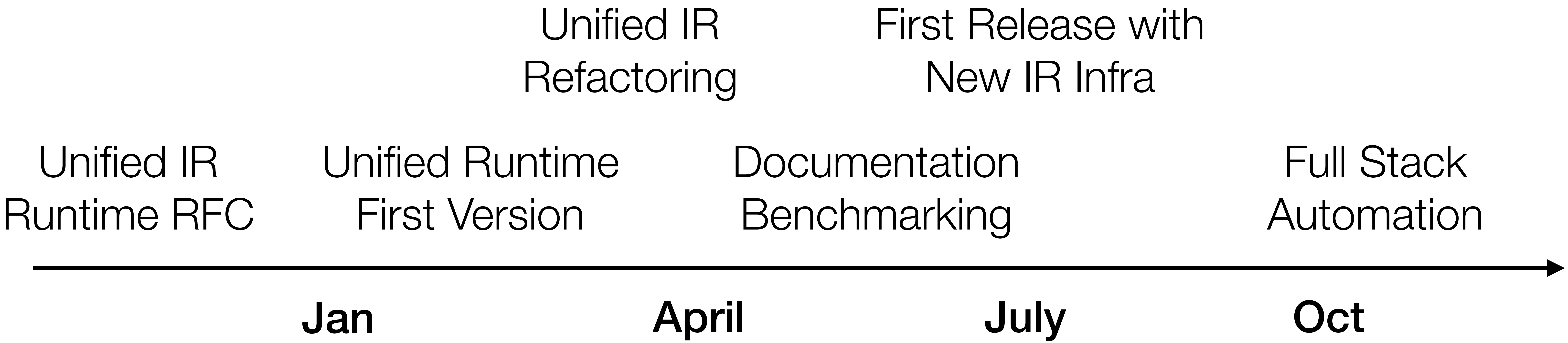
Edge FPGA

Cloud FPGA

ASIC

AutoTVM across
all layers of the stack

2020 Projected Timeline: Selected Topics



Unified IR
Runtime RFC

Unified Runtime
First Version

Unified IR
Refactoring

Documentation
Benchmarking

First Release with
New IR Infra

Full Stack
Automation

Jan

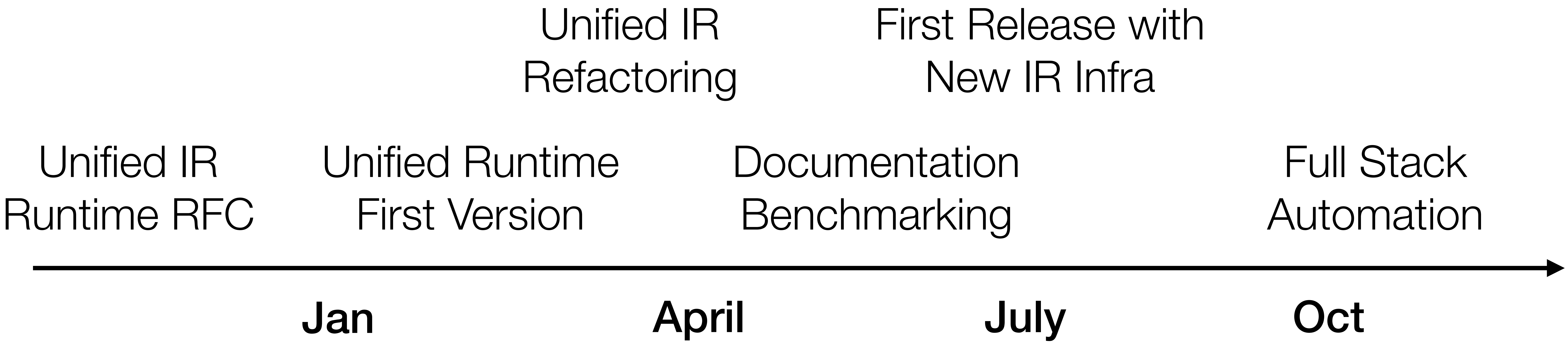
April

July

Oct

2020 Projected Timeline: Selected Topics

Non comprehensive list of on-going topics



Unified IR
Runtime RFC

Unified Runtime
First Version

Unified IR
Refactoring

Documentation
Benchmarking

First Release with
New IR Infra

Full Stack
Automation

Jan

April

July

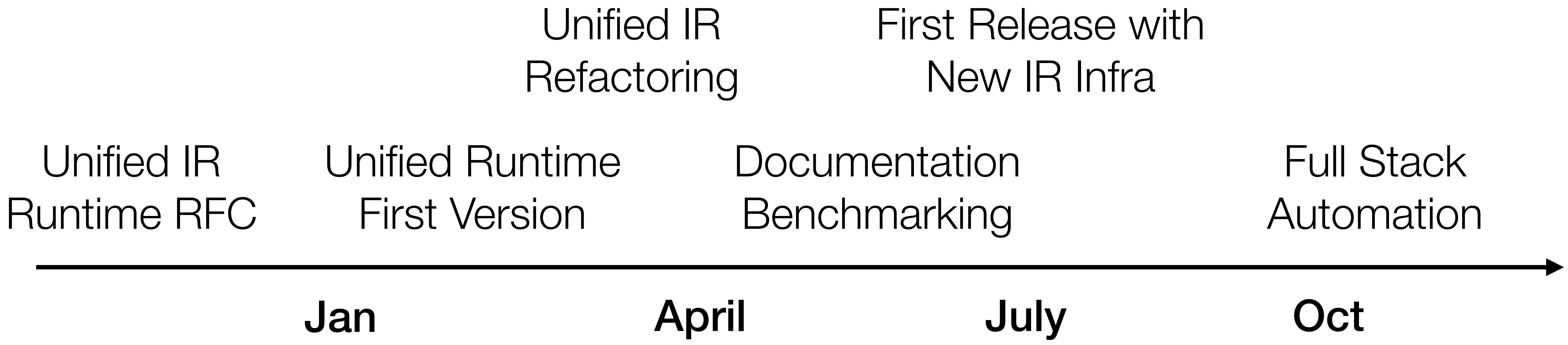
Oct

2020 Projected Timeline: Selected Topics

Non comprehensive list of on-going topics

Ultra Low bits Gradient/Training BERT TSIM AutoSchedule

uTVM Standalone Dynamic Shape NPU coverage



Unified IR Refactoring

First Release with New IR Infra

Unified IR Runtime RFC

Unified Runtime First Version

Documentation Benchmarking

Full Stack Automation

Jan

April

July

Oct

Community

Open Source Community

THE APACHE SOFTWARE FOUNDATION

Incubated as Apache TVM. Independent governance, allowing competitors to collaborate.

Open Source Community

THE APACHE® SOFTWARE FOUNDATION

Incubated as Apache TVM. Independent governance, allowing competitors to collaborate.

Open Source Code

Open Development

Open Governance

Open Source Community

THE APACHE® SOFTWARE FOUNDATION

Incubated as Apache TVM. Independent governance, allowing competitors to collaborate.

Open Source Community

Incubated as Apache TVM. Independent governance, allowing competitors to collaborate.

Growing Developer Community

22 committers, 47 reviewers, 295 contributors

Open Source Community

THE APACHE® SOFTWARE FOUNDATION

Incubated as Apache TVM. Independent governance, allowing competitors to collaborate.

Growing Developer Community

22 committers, 47 reviewers, 295 contributors

~70% growth since TVM Conf 2018

Open Source Community

Incubated as Apache TVM. Independent governance, allowing competitors to collaborate.

Growing Developer Community

22 committers, 47 reviewers, 295 contributors

~70% growth since TVM Conf 2018

Monthly Statistics

~50 authors, ~140 PRs, ~1000 discuss forum posts sampl

tvm.ai

Big THANKS to our sponsors!

HUAWEI

ORACLE®

intel®

XILINX

QUALCOMM®

ada
Applications Driving Architectures

CRISP
Center for Research in Intelligent
Storage and Processing in Memory

SRC®
Semiconductor
Research
Corporation

NSF

DARPA

f

9:00	Keynote & Community Update TVM @ AWS, FB	Keynote (SAMPL, Qualcomm, Amazon, OctoML) TVM @ AWS – Yida Wang, Amazon TVM @ FB — Andrew Tulloch and Bram Wasti, Facebook
11:10	*Break*	
11:30	Compilers and VMs	AI Compilers at Alibaba – Yangqing Jia, Alibaba Dynamic Execution and VMs, Jared Roesch and Haichen Shen, UW and AWS
12:20	Boxed lunches - Contributors Meetup	
13:10	Lightning talks	
13:40	Hardware TVM @ Microsoft, ARM, Xilinx	Building FPGA-Targeted Accelerators with HeteroCL – Zhiru Zhang, Cornell TVM @ Microsoft – Jon Soifer and Minjia Zhang TVM @ ARM – Ramana Radhakrishnan TVM @ Xilinx – Elliott Delaye
15:10	*Break*	
15:30	Automation, new Hardware	TVM @ OctoML – Jason Knight TVM @ Qualcomm – Krzysztof Parzyszek TASO: Optimizing Deep Learning Computation with Automated Generation of Graph Substitutions – Zhihao Jia, Stanford Talk by Nilesh Jain, Intel Labs
16:50	*Break*	
17:00	Lightning talks	
18:10	*Social (food, drinks)*	
20:00	*adjourn*	