ONNX

Open Neural Network Exchange format

Open ecosystem for interchangeable AI models

Dmytro Dzhulgakov
Technical Lead, Engineering Manager
Facebook

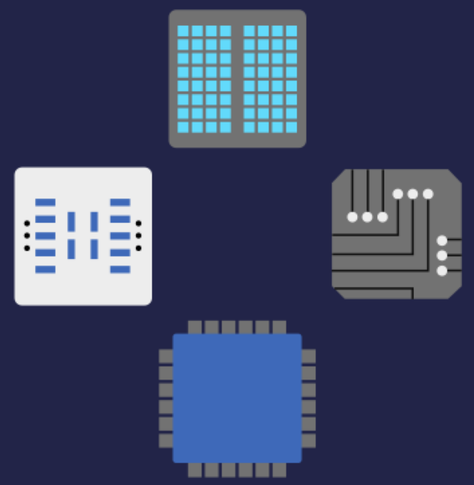
Caffe2

TensorFlow

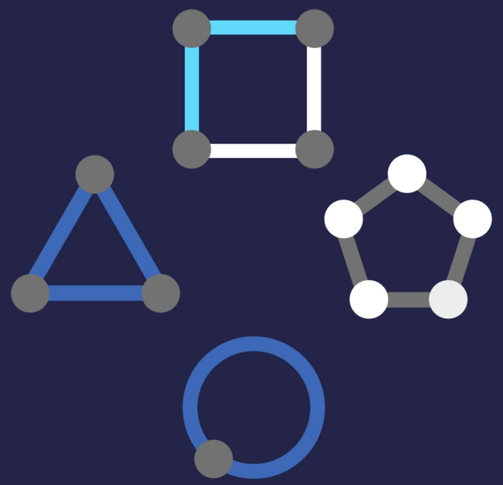
MXNet

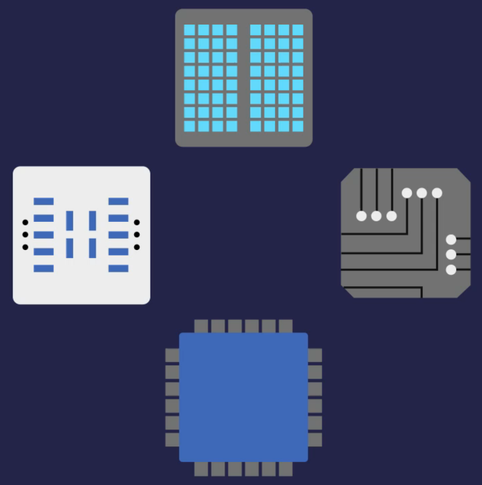
PyTorch

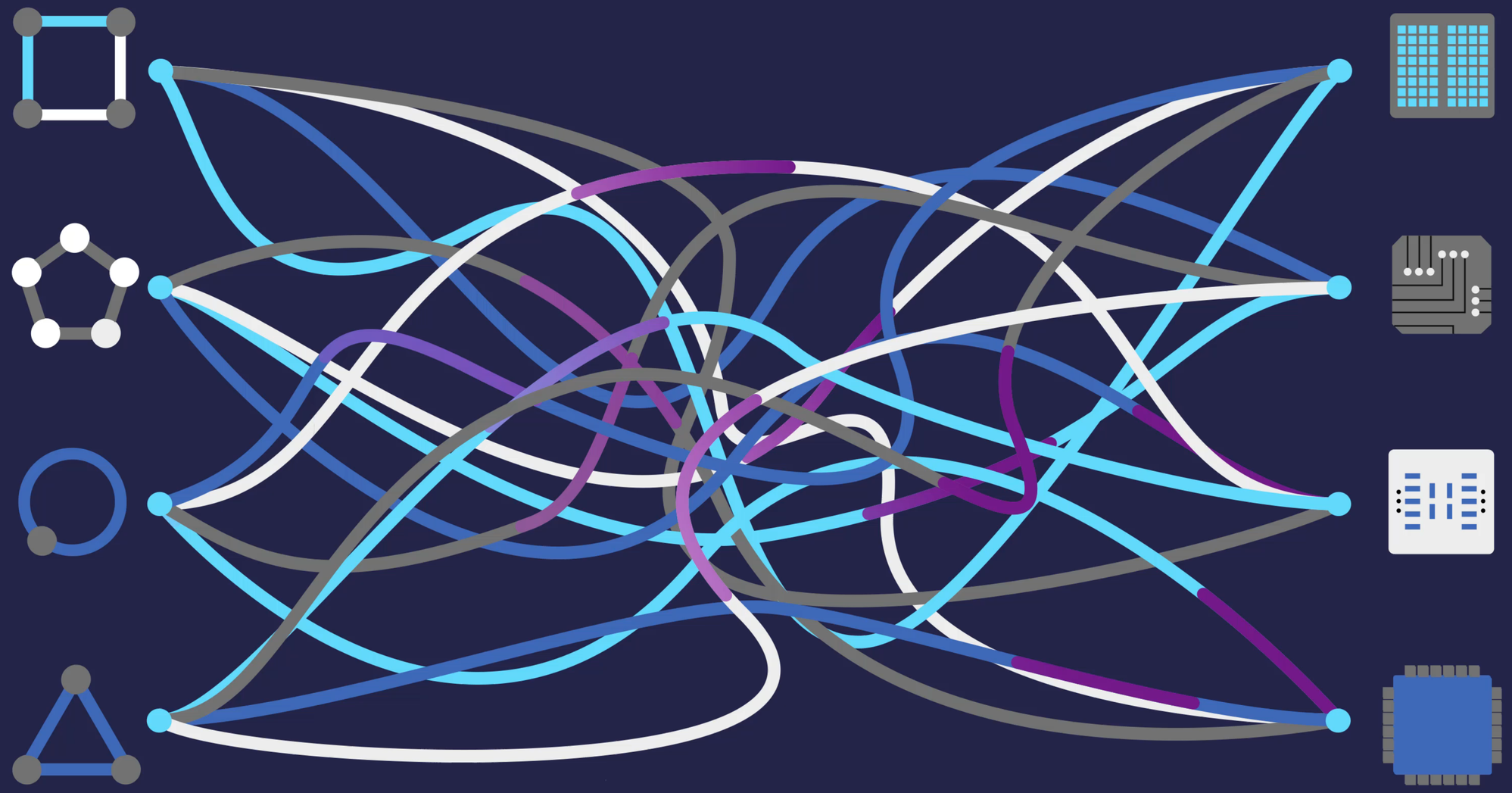
Framework Backends



Hardware-backed libraries
e.g. TensorRT

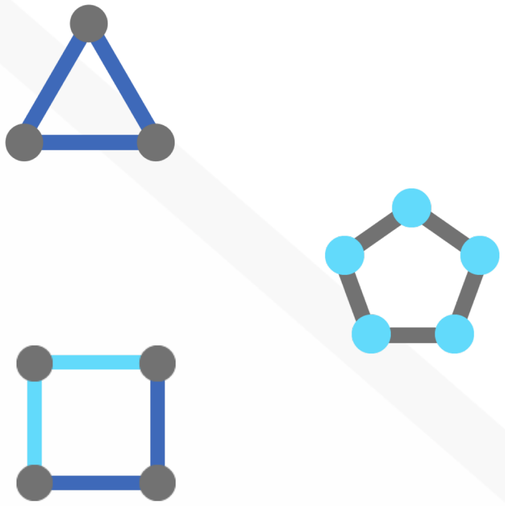




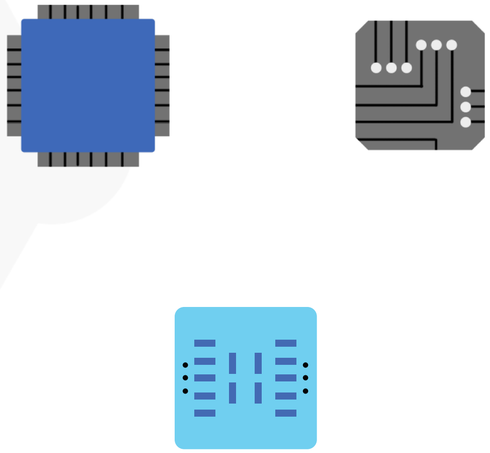


Goals

Provide a standard way to represent models so that:



Serialized models are interoperable between frameworks



Have a common target for optimization for different backends

AI should be OPEN

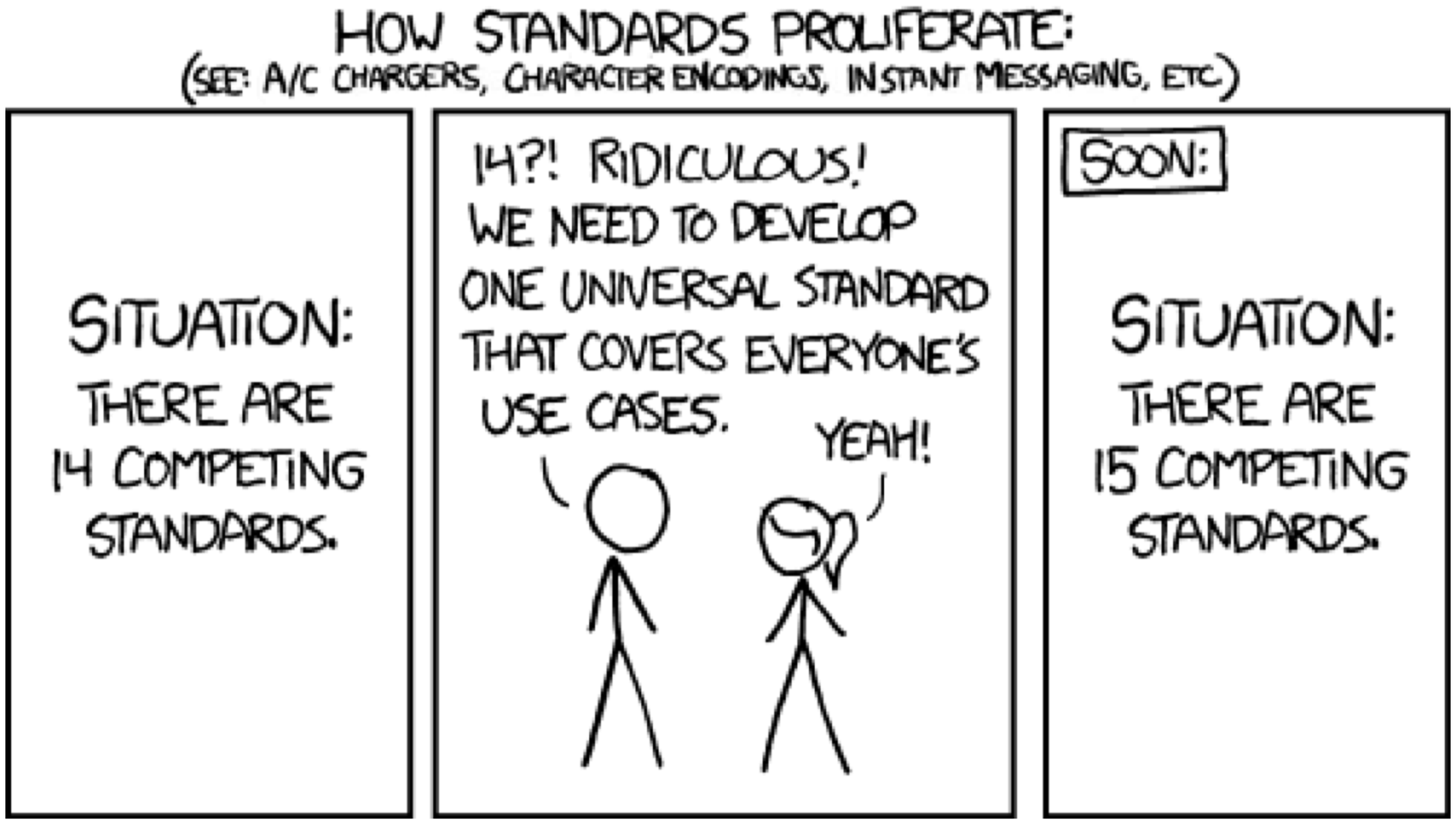
Born from the idea that we should have the freedom to use the best tools

- Framework agnostic
- Open Source & Community Driven
- Better work and better results **together**

ONNX

Open ecosystem
for interoperable
AI models.

Tools Should Work Together.



https://xkcd.com/927/

ONNX Partners

aws

Facebook Open Source

Microsoft

AMD

arm

QUALCOMM

IBM

MEDIATEK

Baidu 百度

Preferred Networks

HUAWEI

NVIDIA

intel

NVidia joins ONNX

NVIDIA + ONNX

News

NVIDIA GPU Cloud Now Available to Hundreds of Thousands of AI Researchers Using NVIDIA Desktop GPUs

Monday, December 4, 2017

Supported Tools

Frameworks

Caffe2

Cognitive Toolkit

mxnet

PYTORCH

Chainer

Converters

ML

Runtimes

NVIDIA

ONNX high-level IR

Initial focus on inference

SSA + Structured control flow

Standard operator definitions

Common rewriting passes

+ set of tests to validate backends

PRelu

PRelu takes input data (Tensor) and slope tensor as input, and produces one output data (Tensor
`f(x) = slope * x for x < 0`, `f(x) = x for x >= 0`., is applied to the data tensor elementwise.

Inputs

`X` : *T*
Input tensor

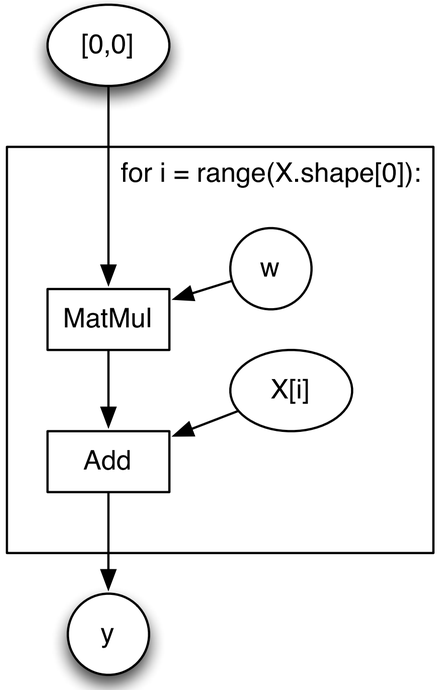
`Slope` : *T*
Slope tensor. If `Slope` is of size 1, the value is sharedacross different channels

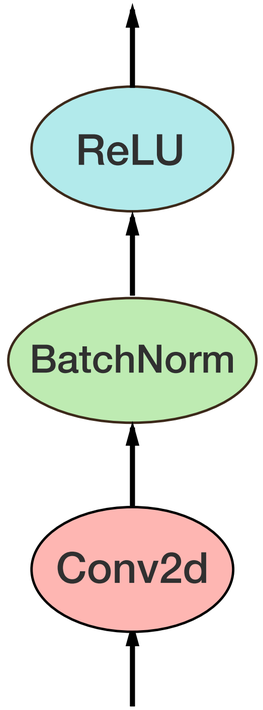
Outputs

`Y` : *T*
Output tensor

Type Constraints

`T` : *tensor(float16), tensor(float), tensor(double)*
Constrain input and output types to float tensors.

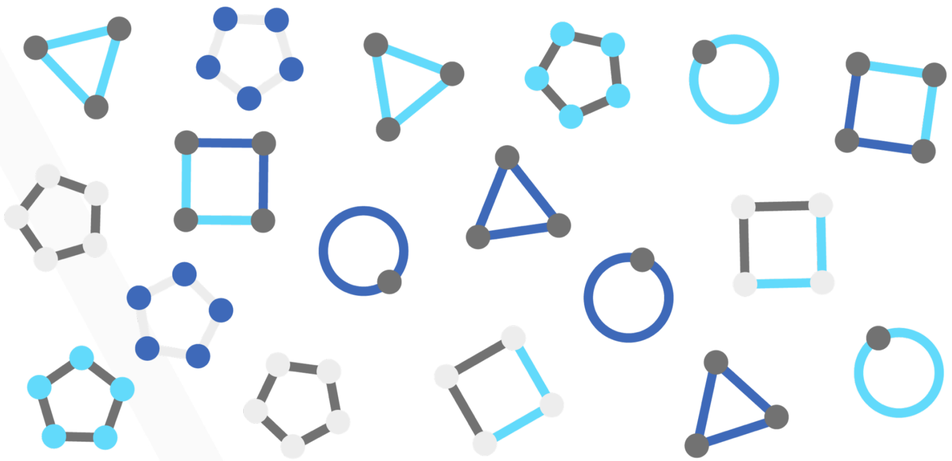


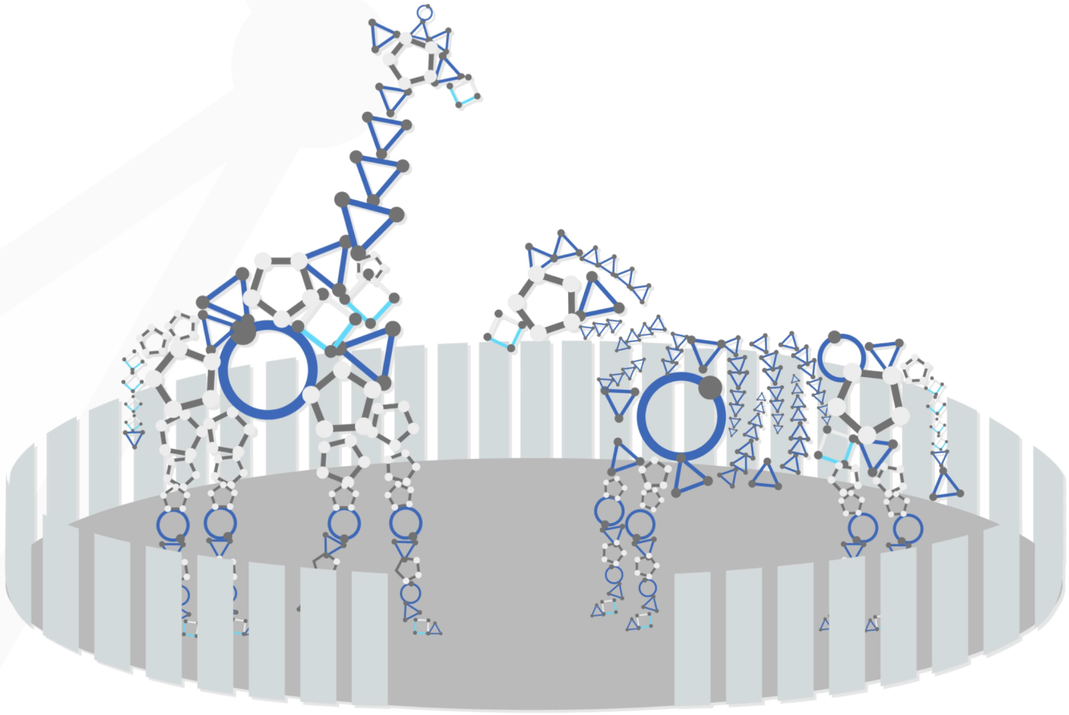


ONNX Model zoo

Collection of popular models

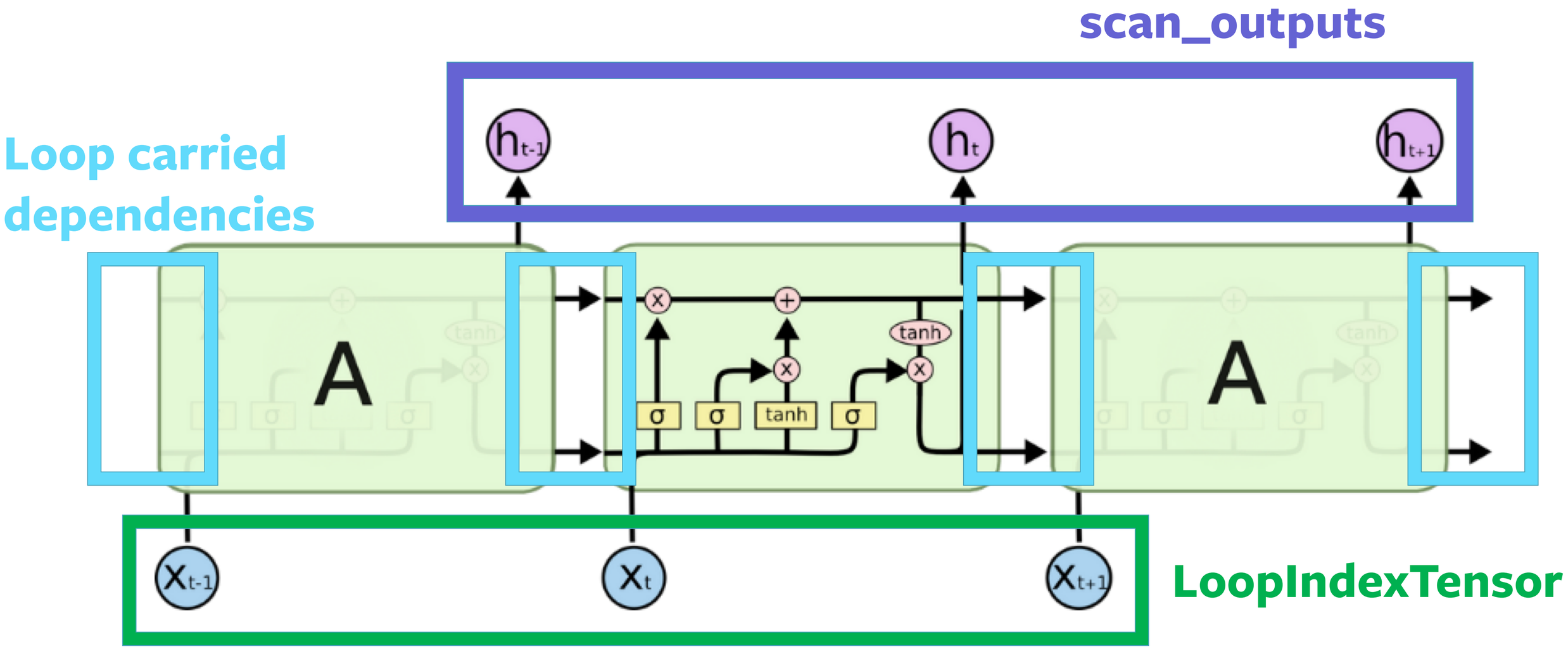
- Continuous testing against frameworks and backends
- Target for optimizations
- Useful as pre-trained model collection





Control Flow

Loop and **If** ops, optimized for NN patterns



Quantization support

Representation for low-precision (in progress):

- float16
- int8, int16
- representing scale/bias for quantization

ONNX

ONNX V1.0 is READY

stable for vision applications
beta for NLP

Research to Production at Facebook

PYTORCH

Reimplementation
takes weeks or months

Caffe2

Research to Production at Facebook

PYTORCH ⟷ ONNX ⟷ Caffe2

Enabling model or model fragment transfer

PYTORCH

define-by-run

```
def foo(x, t):
  y = x.mm(x)
  print(y) # still works!
  return y + t

x = torch.Tensor([[1,2],[3,4]])
foo(x, 1)
```

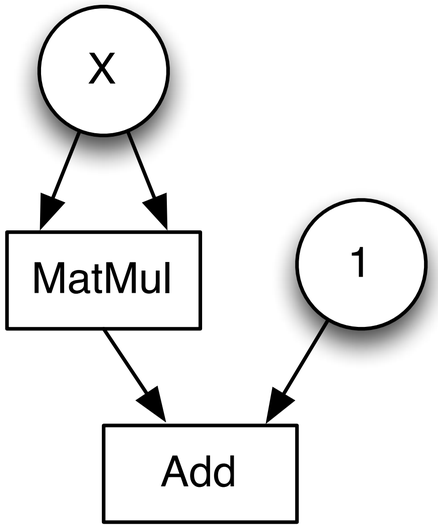
PYTORCH → Serialized model

define-by-run

```
def foo(x, t):
  y = x.mm(x)
  print(y) # still works!
  return y + t

x = torch.Tensor([[1,2],[3,4]])
foo(x, 1)

onnx.export(foo, (x, 1), “my.onnx”)
```



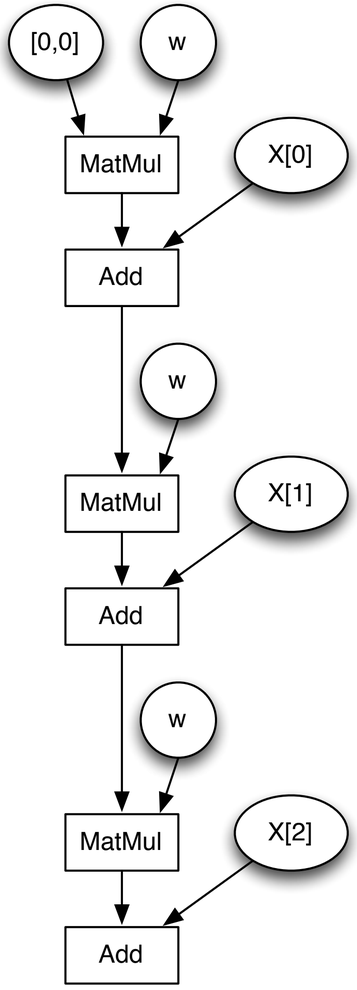
PYTORCH → Serialized model

define-by-run

```
def foo(y, w, t):
  y = y.mm(w)
  print(y) # still works!
  return y + t


def bar(x, w):
    y = torch.zeros(1, 2)
    for t in x:
        y = foo(y, w, t)
    return y

onnx.export(bar, (x, w), “my.onnx”)
```



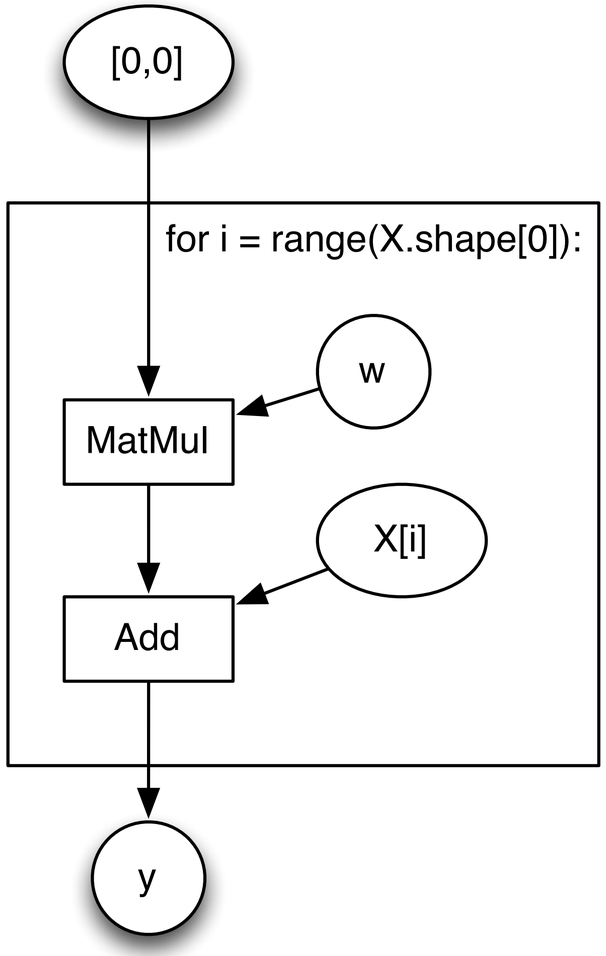
PYTORCH → Serialized model

define-by-run

```
@traced
def foo(y, w, t):
  y = y.mm(w)
  print(y) # still works!
  return y + t

@script
def bar(x, w):
    y = torch.zeros(1, 2)
    for t in x:
        y = foo(y, w, t)
    return y

onnx.export(bar, (x, w), “my.onnx”)
```



NVidia TensorRT

TensorRT – optimized engine for Nvidia GPUs

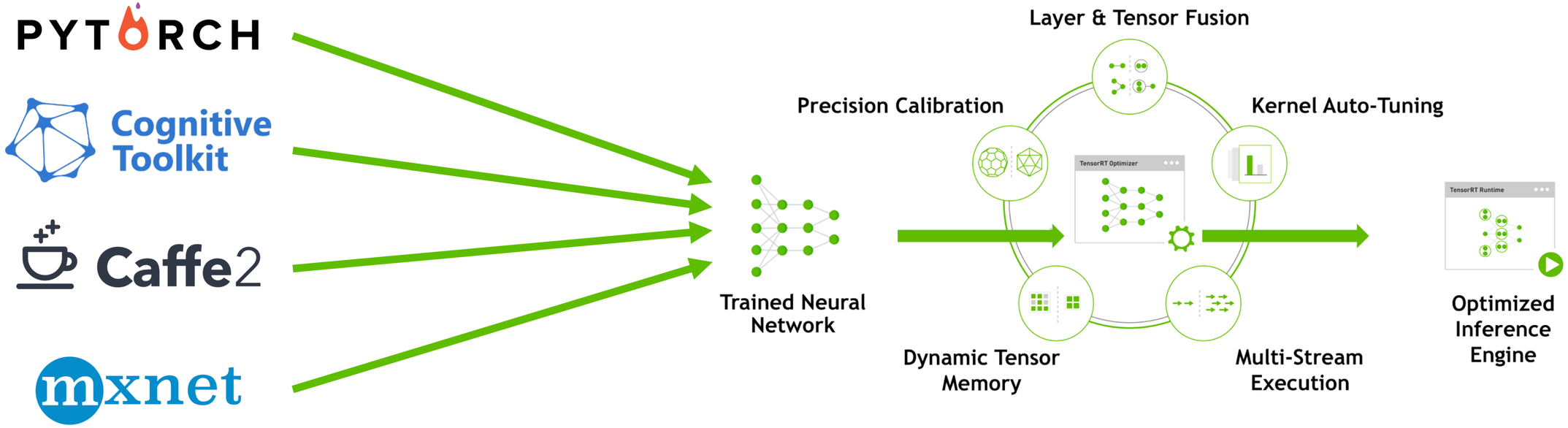


Image credit: https://devblogs.nvidia.com/tensorrt-3-faster-tensorflow-inference/

NVidia TensorRT + ONNX

ONNX as a model ingestion layer

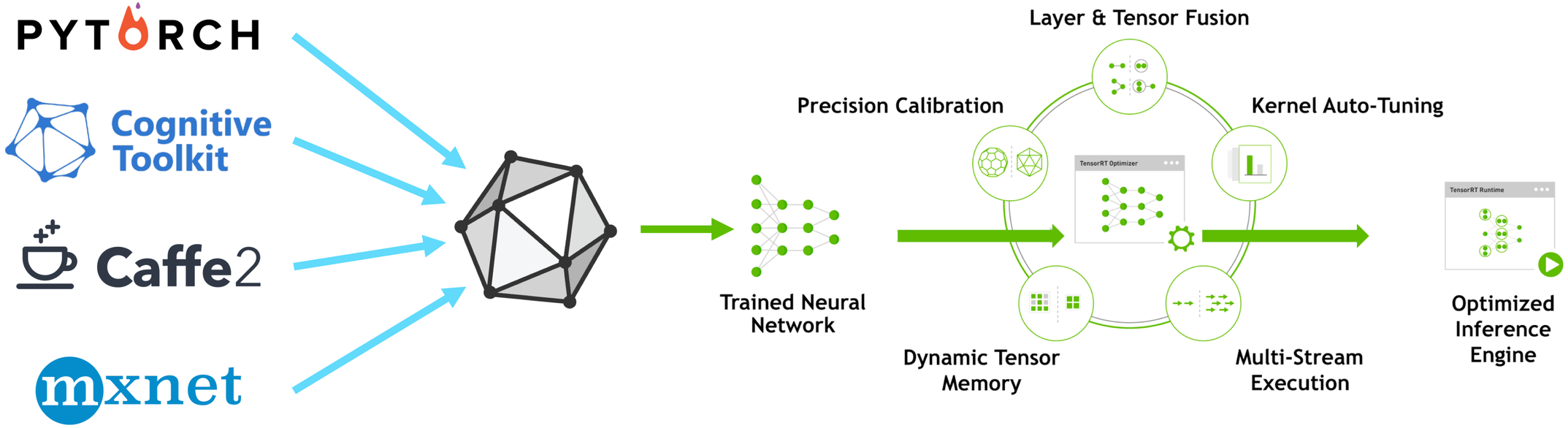
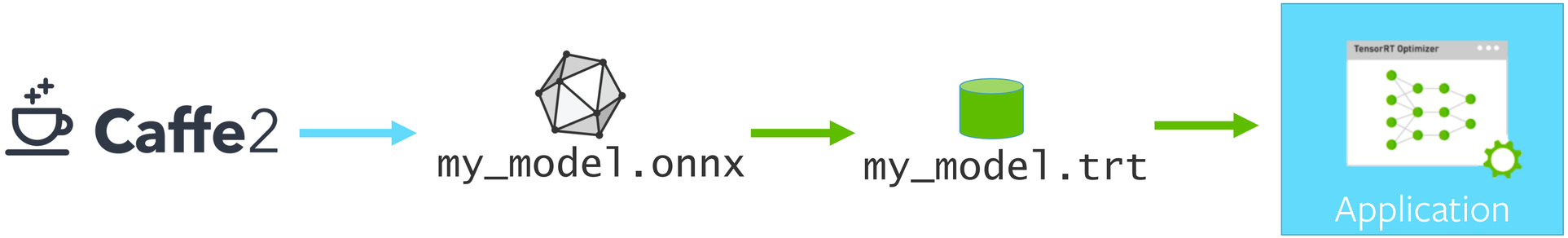


Image credit: https://devblogs.nvidia.com/tensorrt-3-faster-tensorflow-inference/

Ahead of time conversion

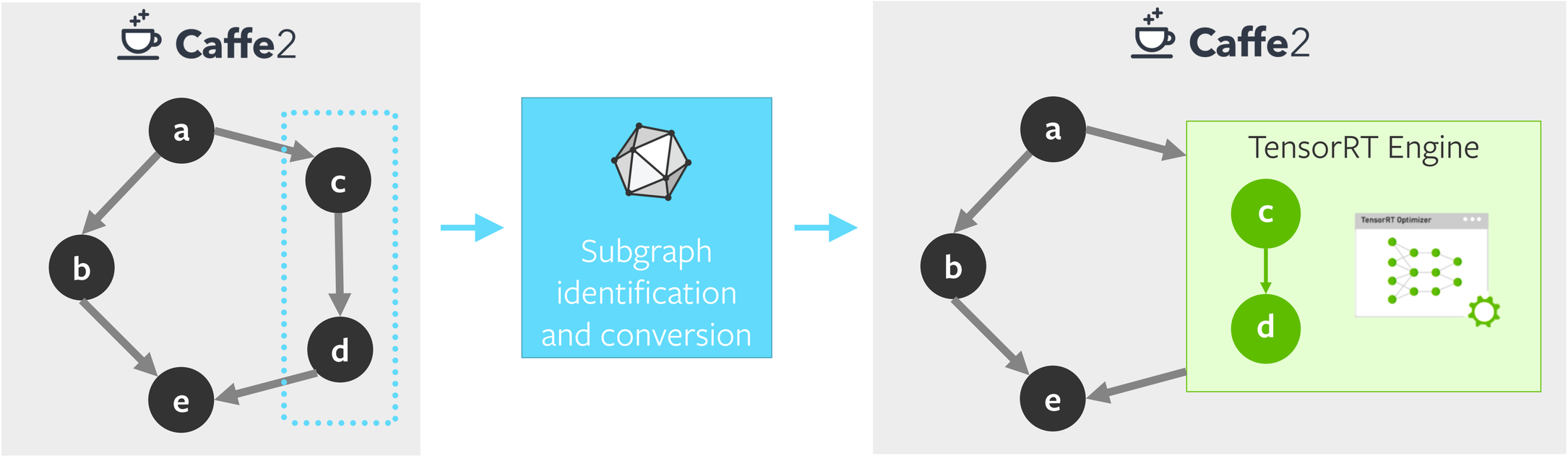
Convert entire model to TensorRT file format

```
$ onnx2trt my_model.onnx -o my_engine.trt
```



Realtime conversion

Use C API to invoke conversion inside the framework



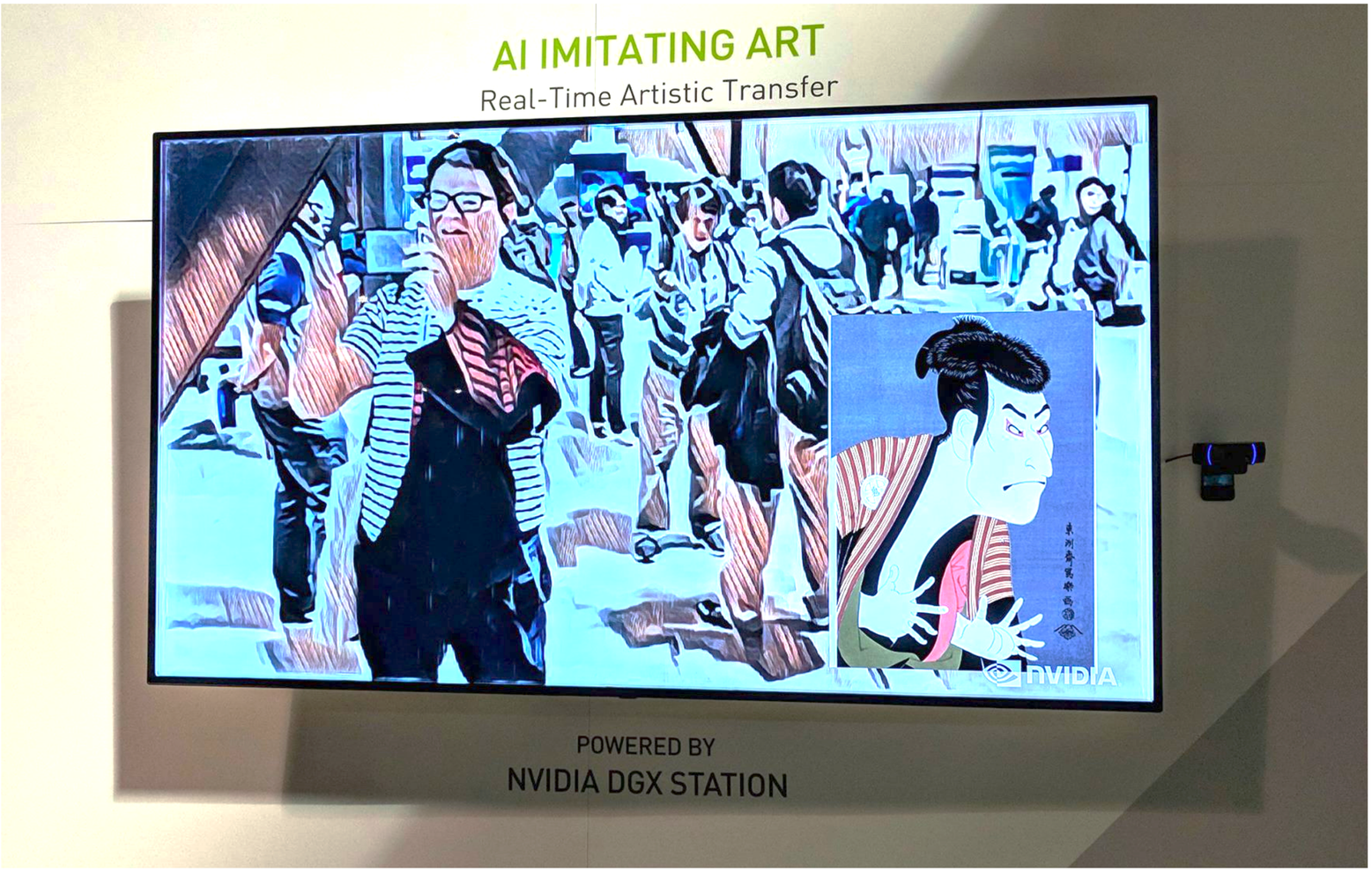
Demo at NIPS 2017

Trained in **PyTorch**

Exported to **ONNX**

Running on **DGX Station**

Powered by **TensorRT**



ONNX + TensorRT

+ Caffe2 on-the-fly integration

ONNX is a Community Project

Get Involved

Contribute

ONNX is a community project. We encourage you to join the effort and contribute feedback, ideas, and code. Join us on Github.

github.com/onnx

Use ONNX

Start experimenting today. Check out our Getting Started Guide, Supported Tools, and Tutorials.

ONNX.ai

Follow Us

Stay up to date with the latest ONNX news.

onnxai

onnxai

Join the **Working Groups**

onnx@onnx.ai

Collaborations across cloud providers, hardware vendors and end users:

- Quantization
- RNNs and Control Flow
- Test and Compliance
- Model Training

ONNX

Questions?