Learning to Optimize Tensor Programs

Tianqi Chen, Lianmin Zheng, Eddie Yan, Ziheng Jiang, Thierry Moreau,
Luis Ceze, Carlos Guestrin, Arvind Krishnamurthy

PAUL G. ALLEN SCHOOL
OF COMPUTER SCIENCE & ENGINEERING

tvm.ai

sampl

Goal: Deploy Deep Learning Everywhere

Frameworks

Explosion of models and frameworks

Huge gap between model/frameworks and hardware backends

Explosion of hardware backends

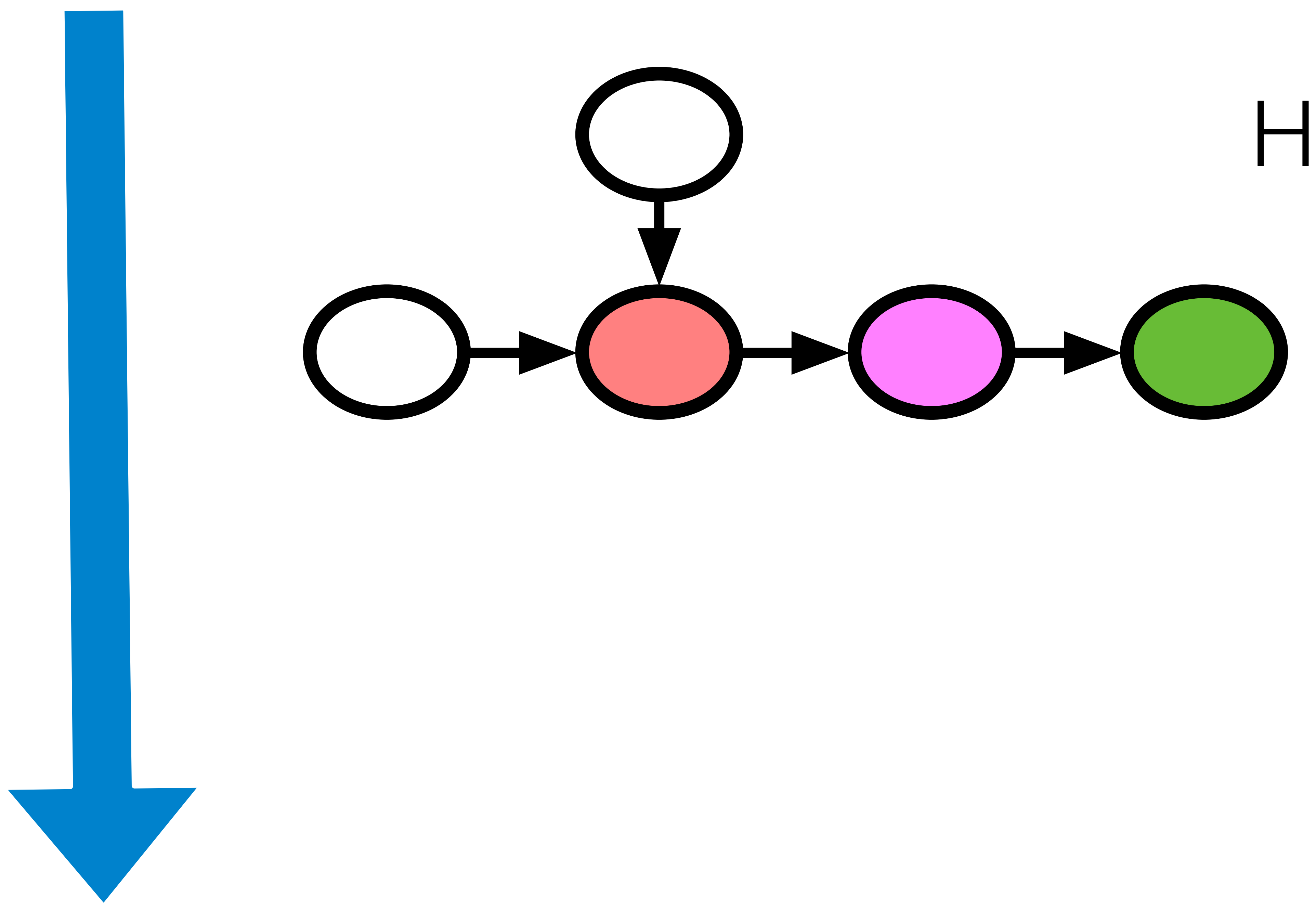
Existing Approach

Frameworks

Hardware

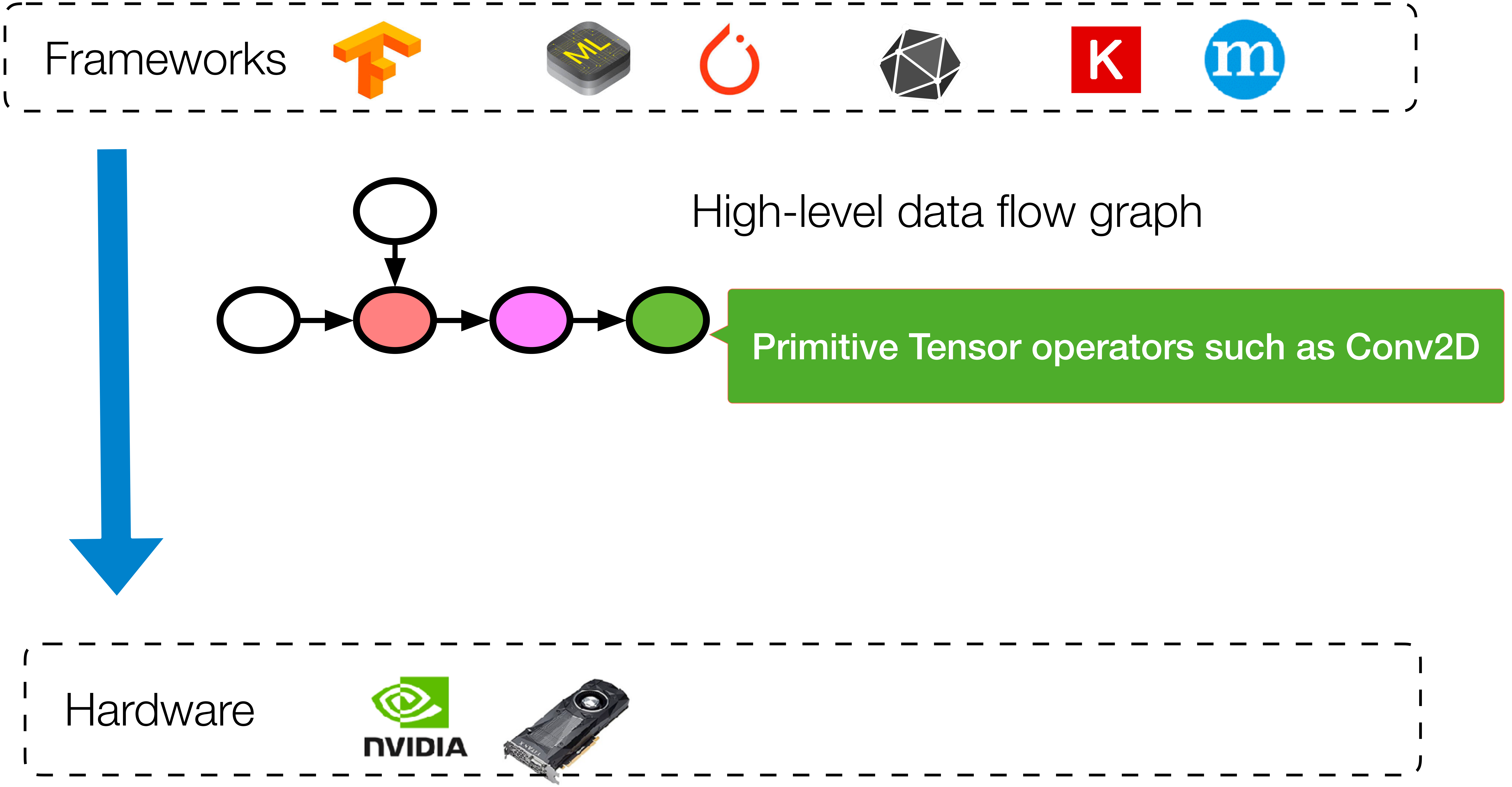
Existing Approach

Frameworks

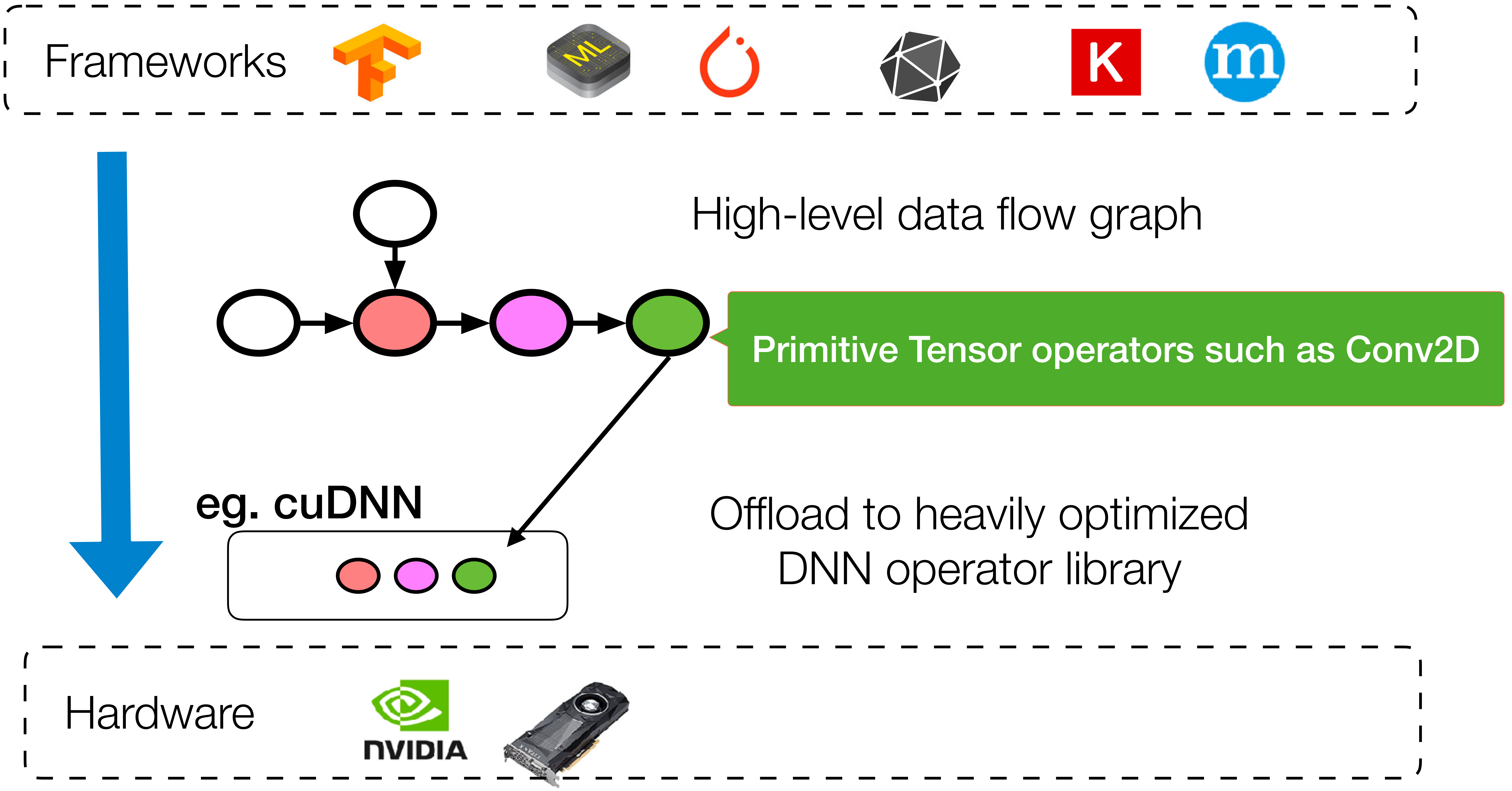


Hardware

Existing Approach

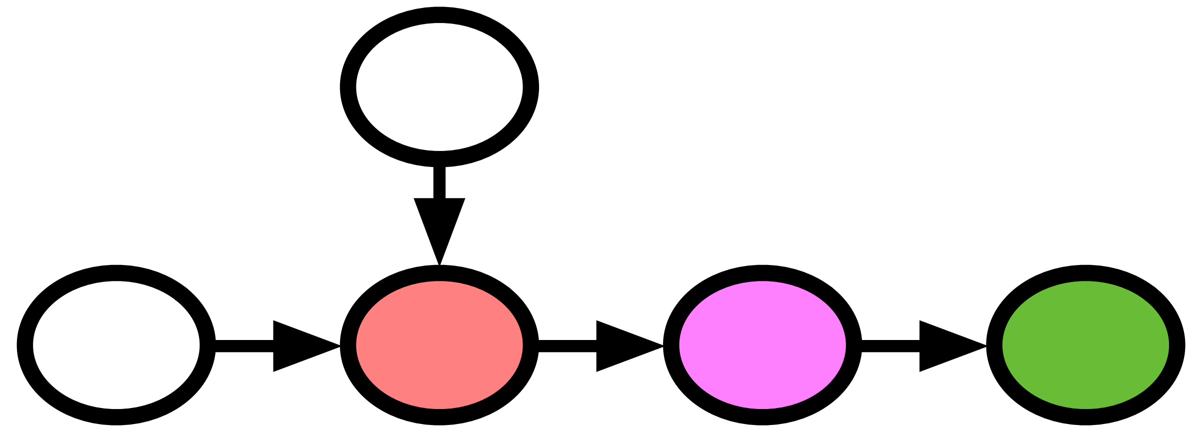

Frameworks
High-level data flow graph
Primitive Tensor operators such as Conv2D
Hardware
NVIDIA

Existing Approach


Frameworks
High-level data flow graph
Primitive Tensor operators such as Conv2D
eg. cuDNN
Offload to heavily optimized DNN operator library
Hardware
NVIDIA

Limitations of Existing Approach

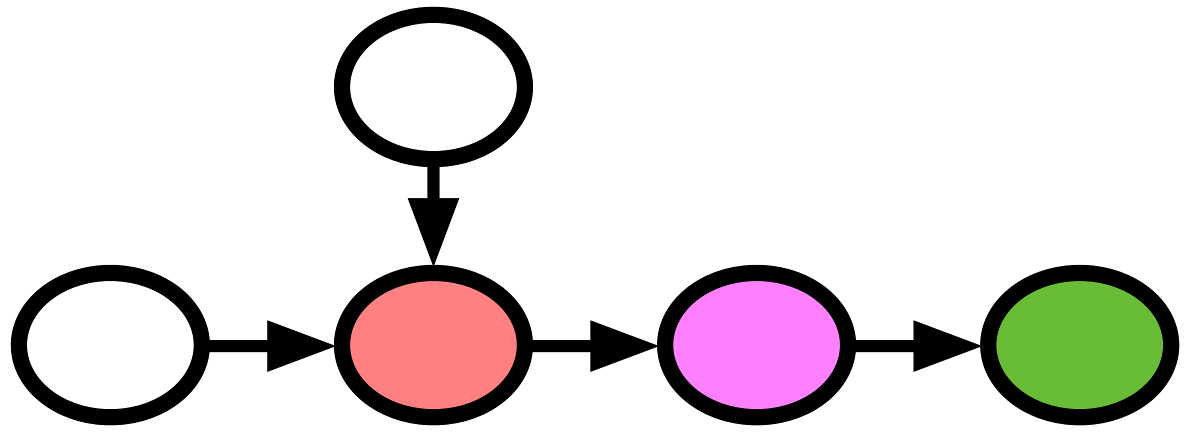
Frameworks



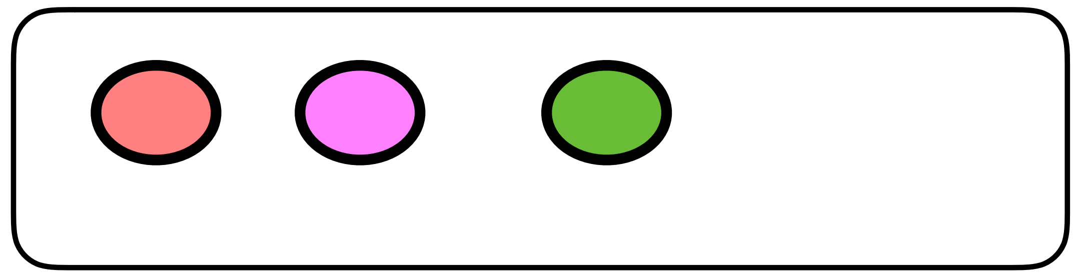
cuDNN

Limitations of Existing Approach

Frameworks

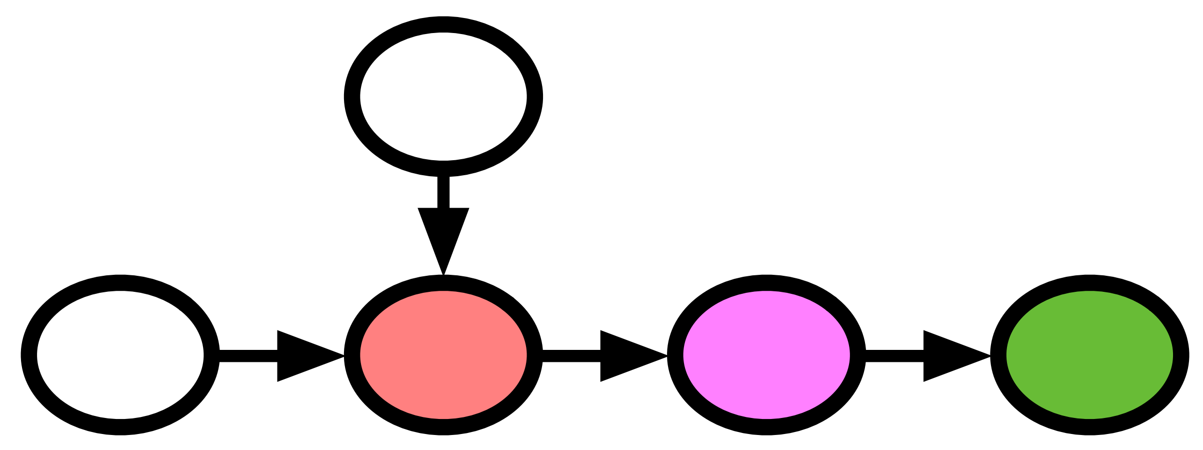


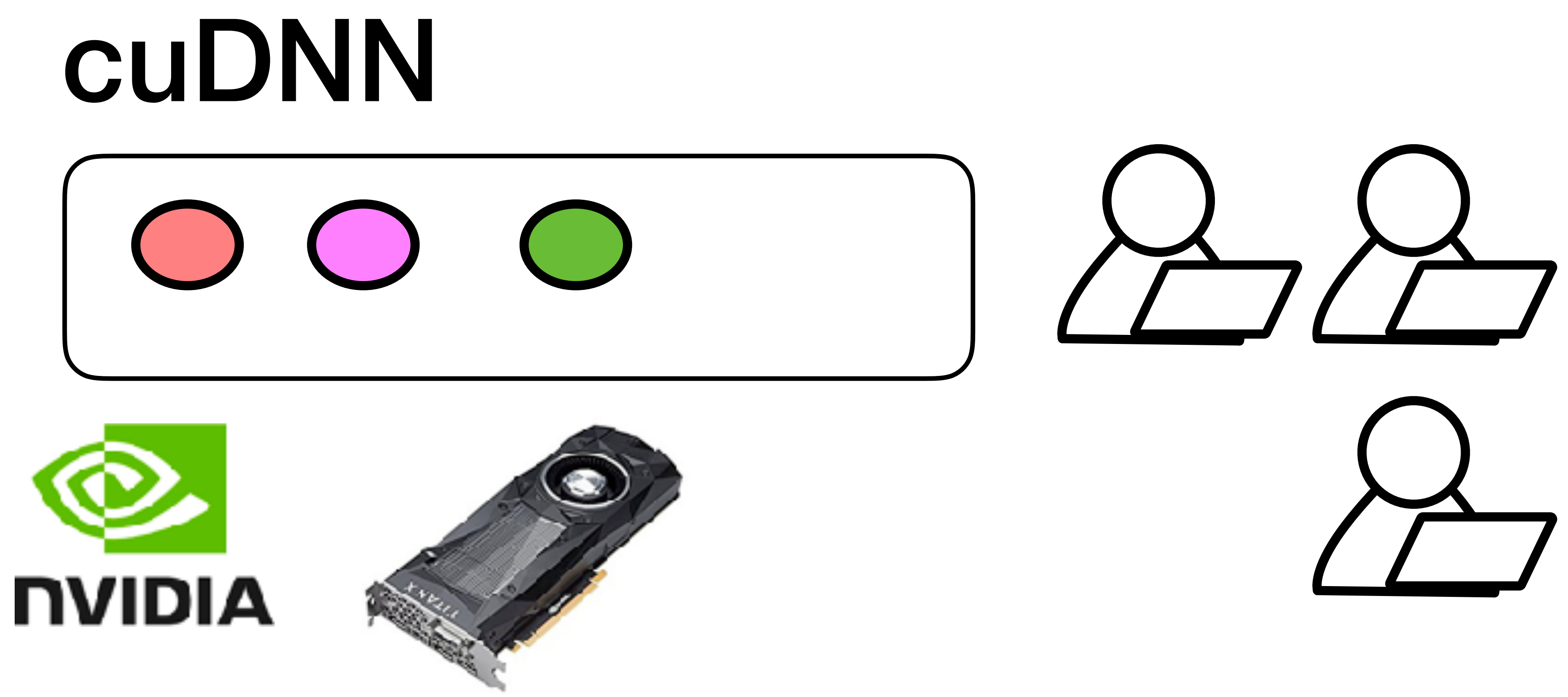
cuDNN



Limitations of Existing Approach

Frameworks

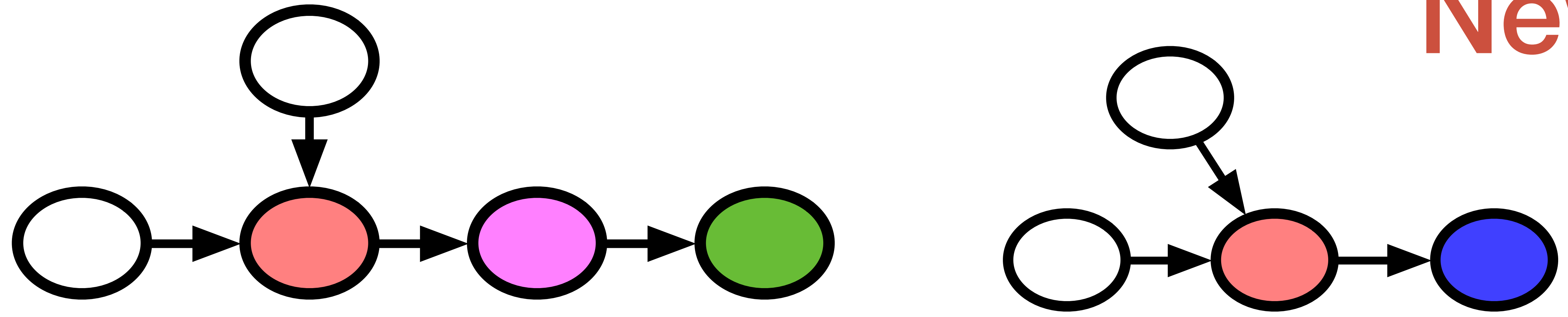




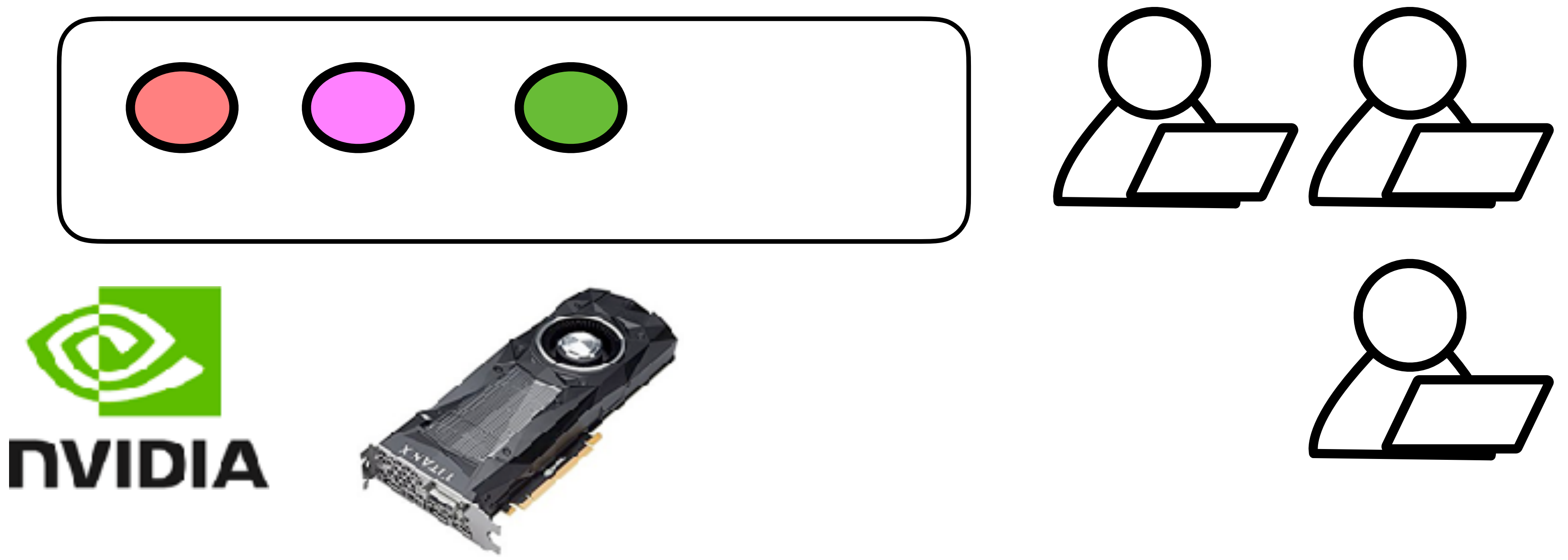
Limitations of Existing Approach

Frameworks

New operators



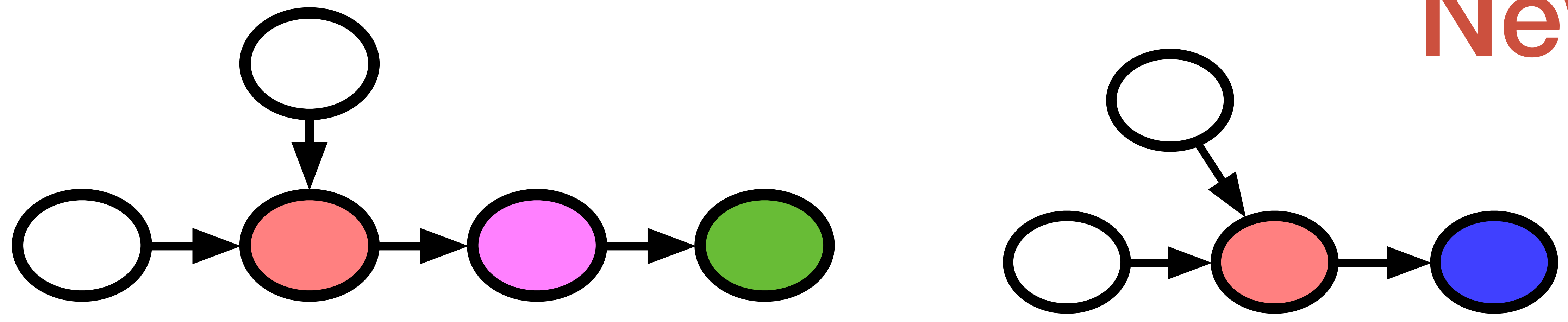
cuDNN



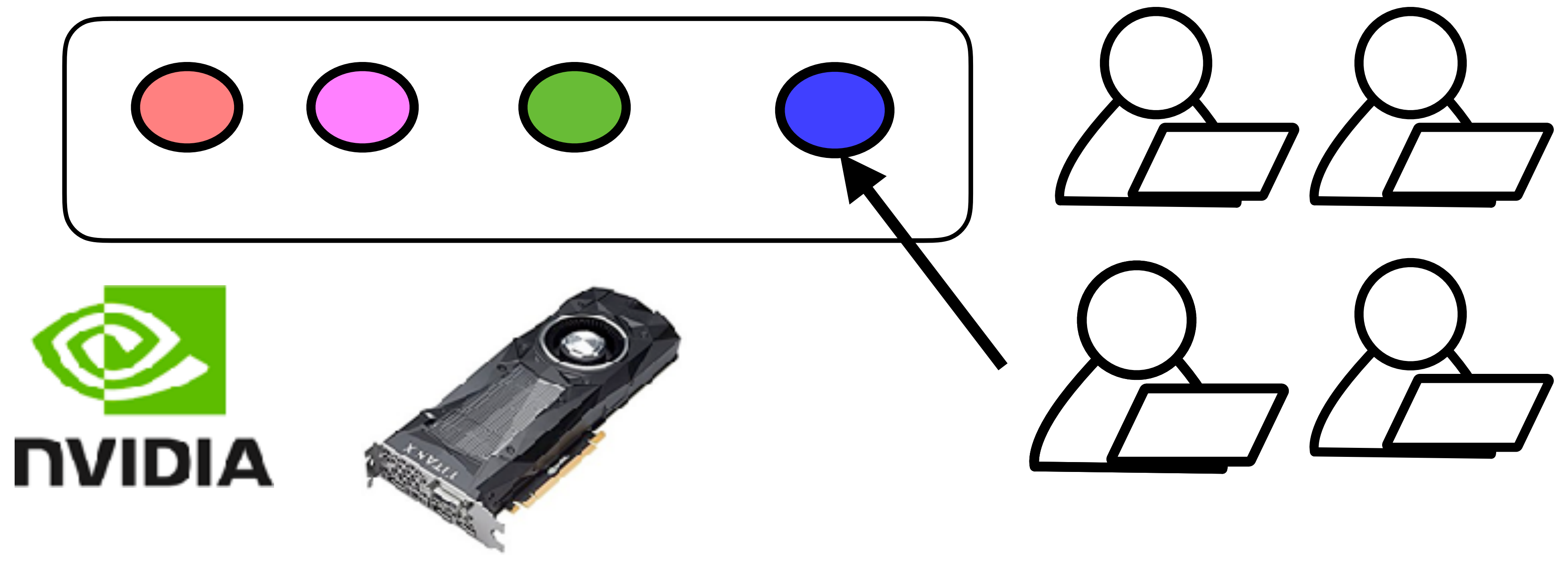
Limitations of Existing Approach

Frameworks

New operators

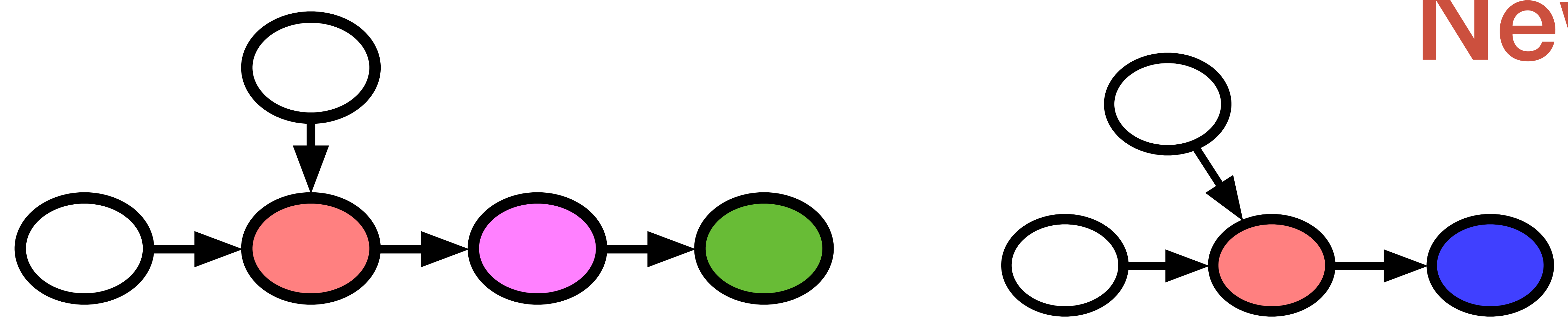


cuDNN



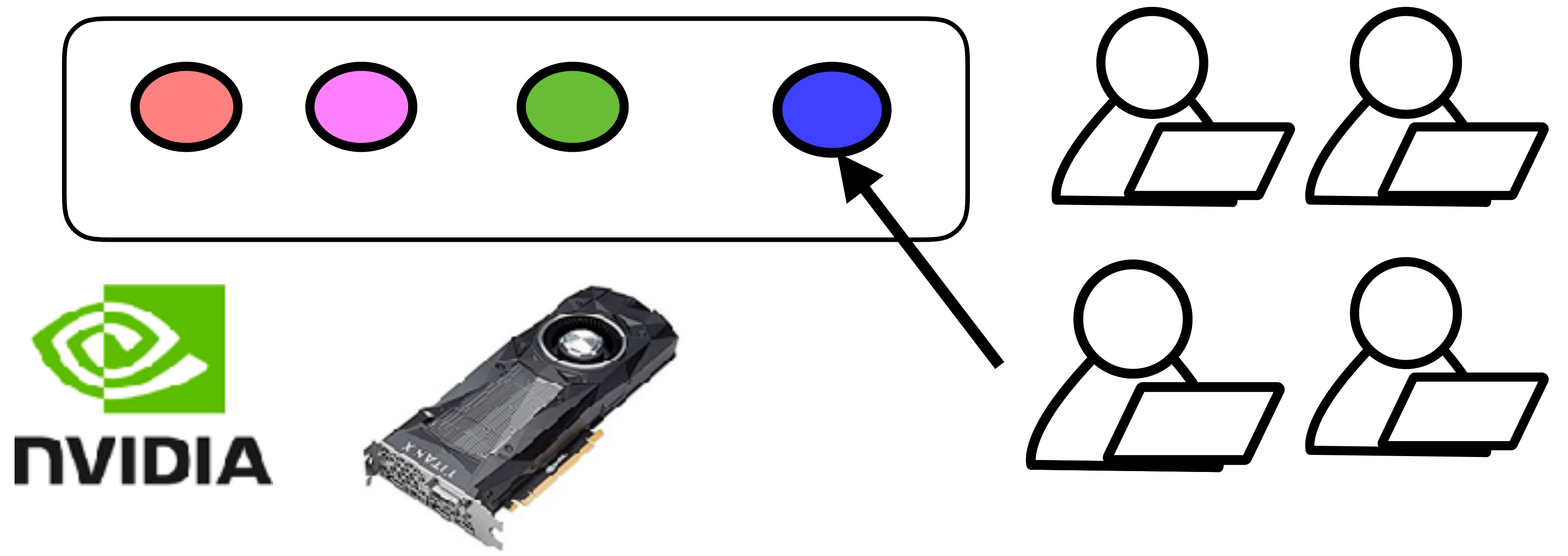
Limitations of Existing Approach

Frameworks



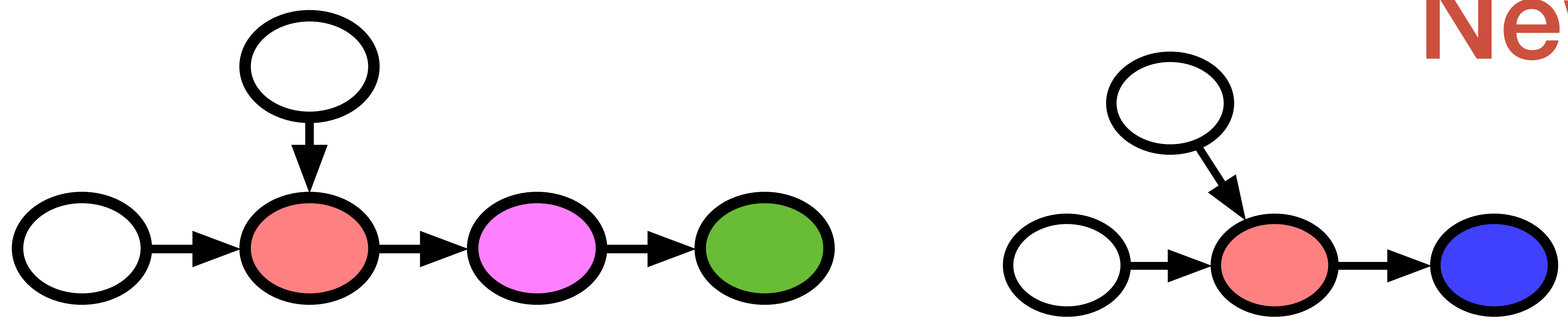
New operators

cuDNN



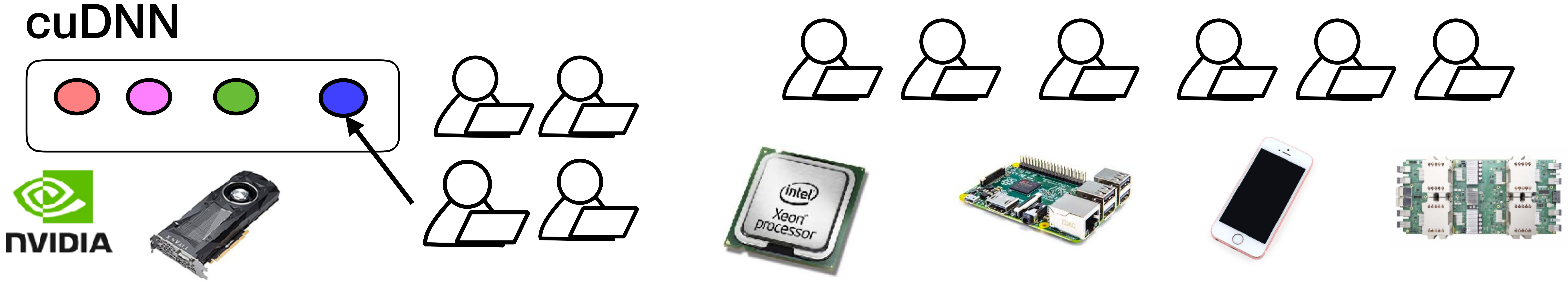
Limitations of Existing Approach

Frameworks

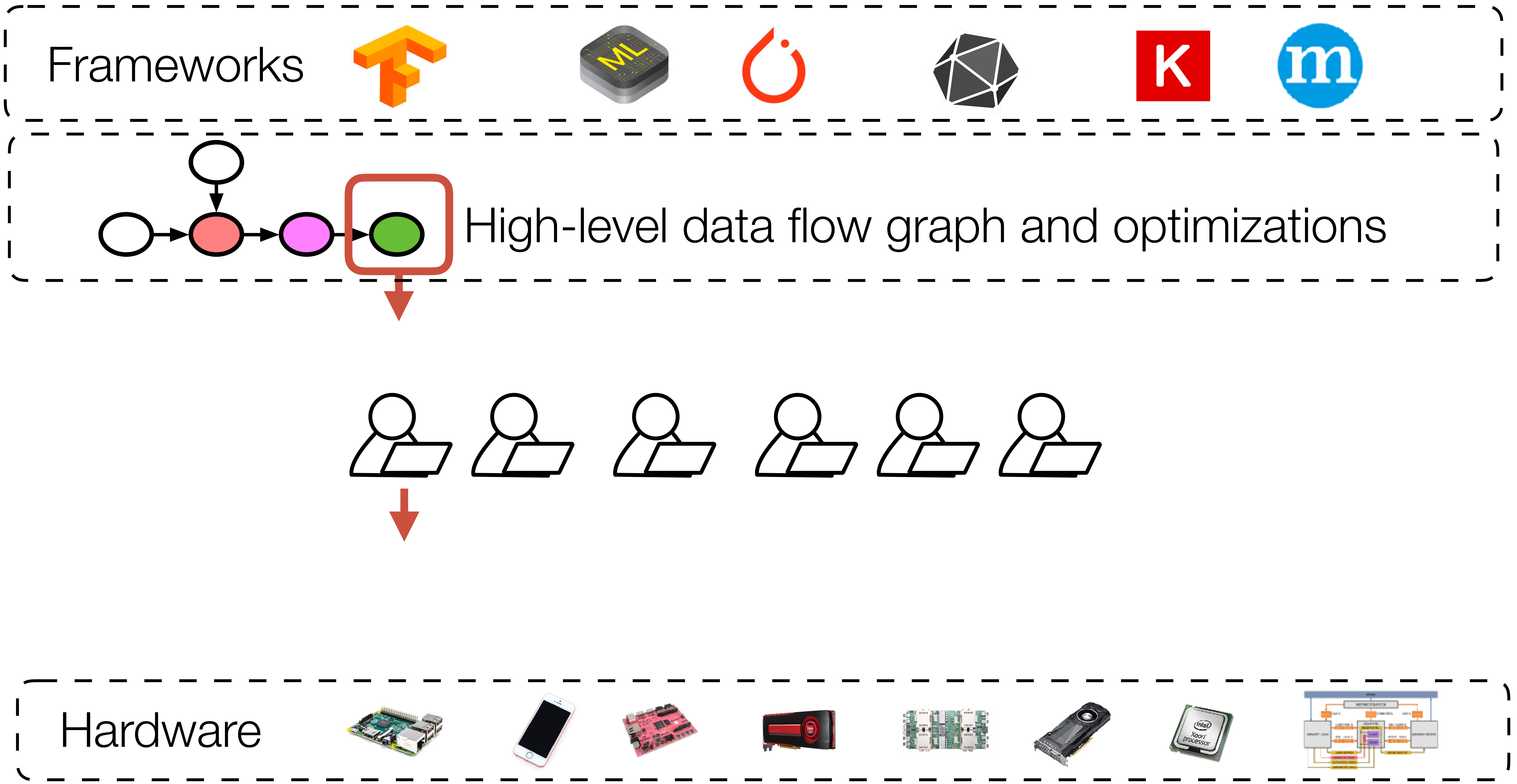


New operators

Engineering intensive

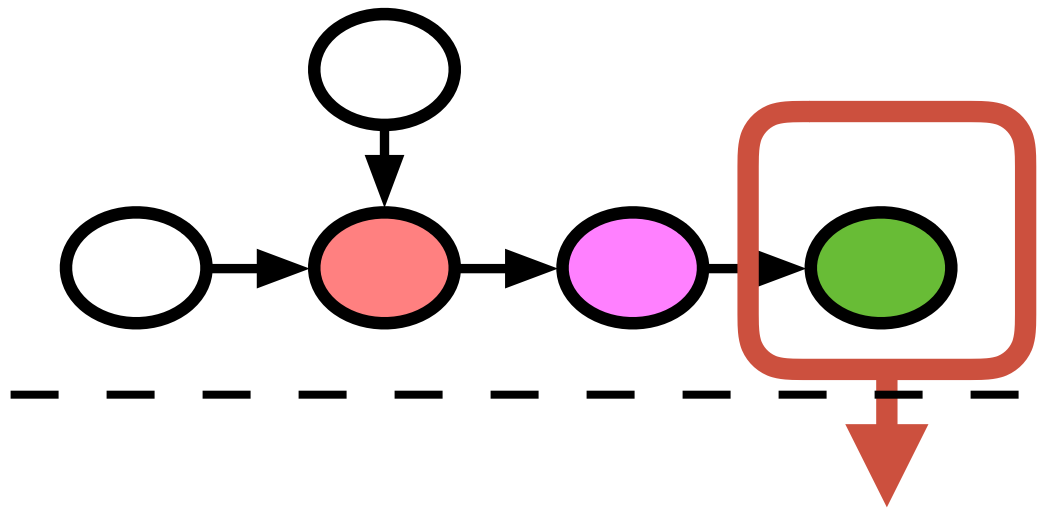


Learning to Optimize Tensor Programs



Learning to Optimize Tensor Programs

Frameworks

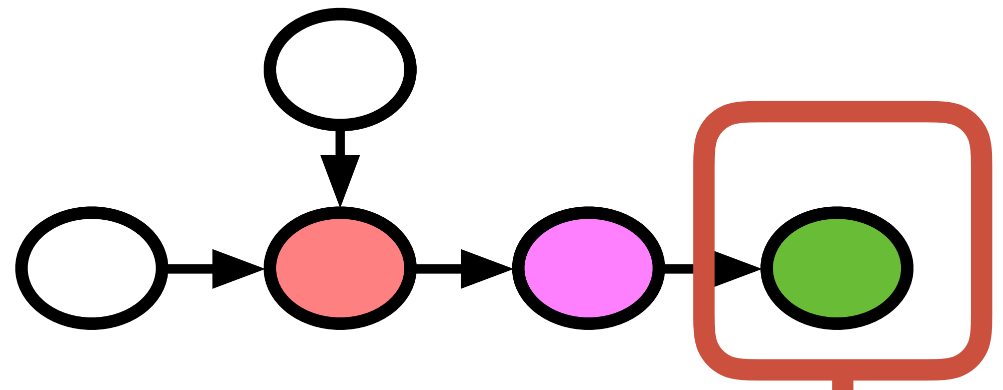


High-level data flow graph and optimizations

Hardware

Learning to Optimize Tensor Programs

Frameworks



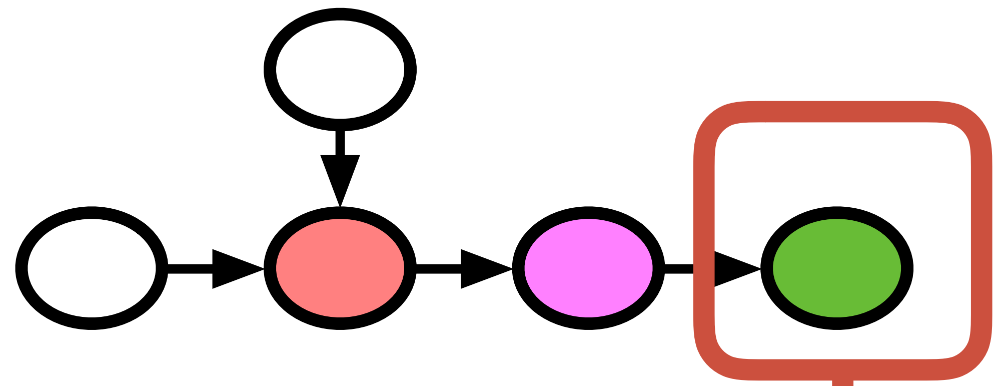
High-level data flow graph and optimizations

Machine Learning based Program Optimizer

Hardware

Learning to Optimize Tensor Programs

Frameworks

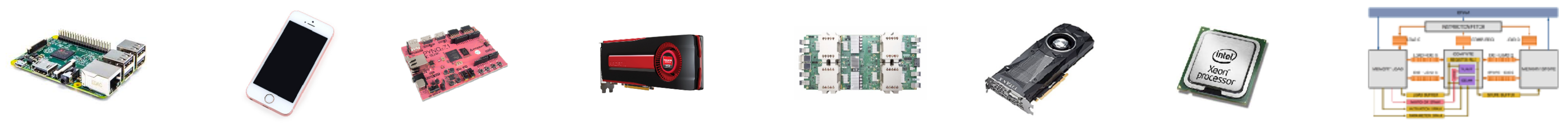


High-level data flow graph and optimizations

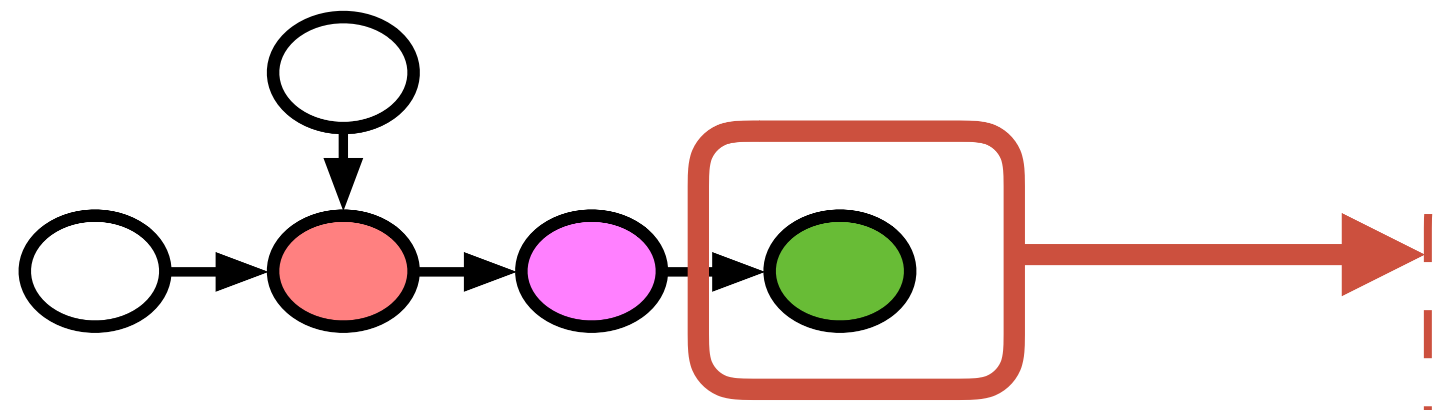
Machine Learning based Program Optimizer

Learning to generate optimized program
for new operator workloads and hardware

Hardware



Search over Possible Program Transformations



Compute Description

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

Loop Transformations

Thread Bindings

Cache Locality

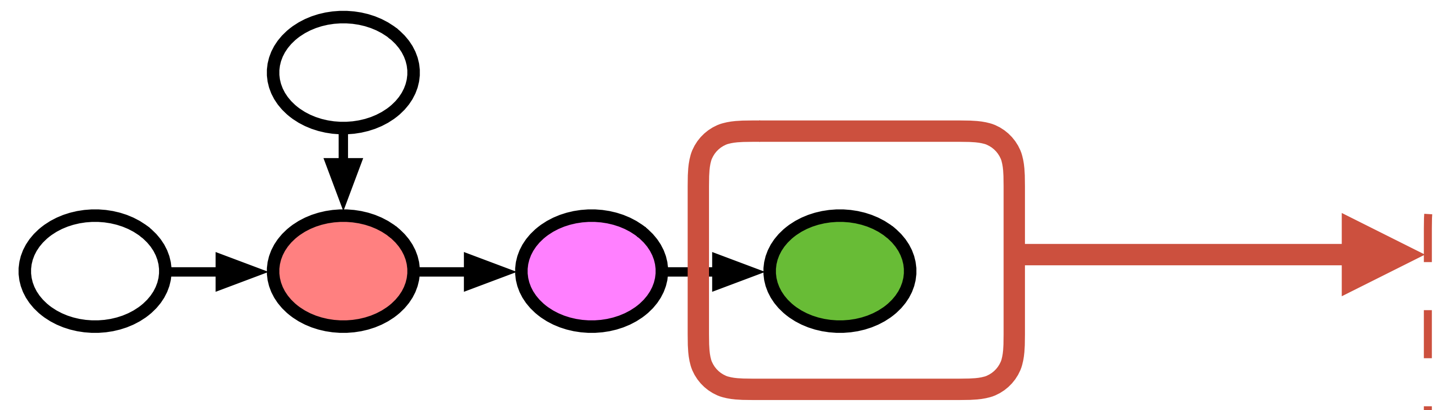
Thread Cooperation

Tensorization

Latency Hiding

Hardware

Search over Possible Program Transformations



Compute Description

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

Loop Transformations

Thread Bindings

Cache Locality

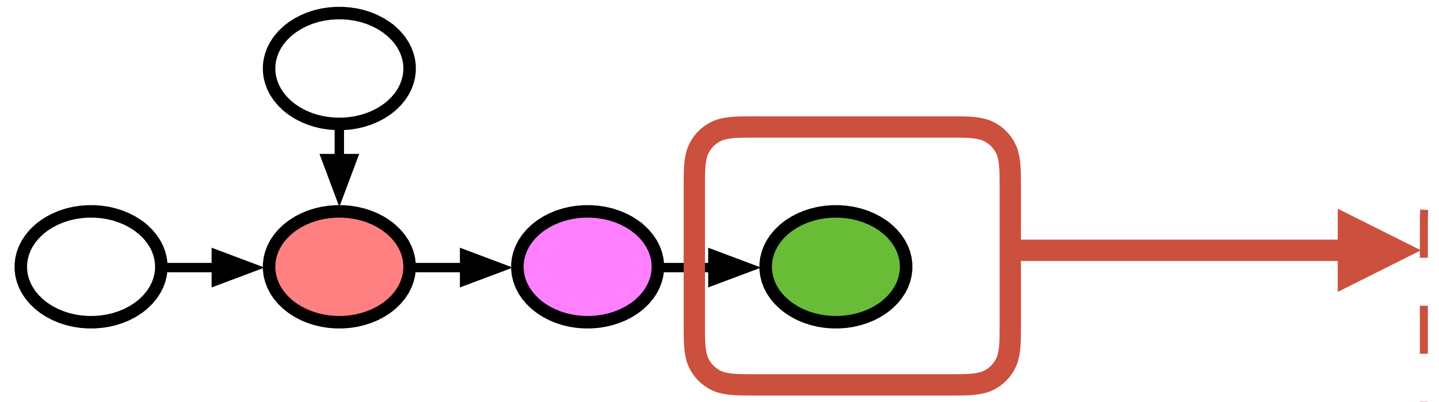
Thread Cooperation

Tensorization

Latency Hiding

Hardware

Search over Possible Program Transformations



Compute Description

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

Billions of possible optimization choices

Loop Transformations

Thread Bindings

Cache Locality

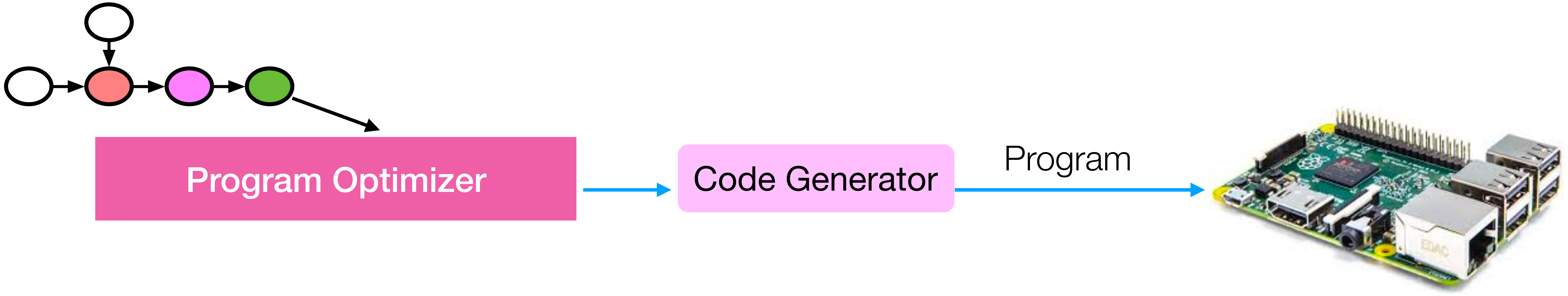
Thread Cooperation

Tensorization

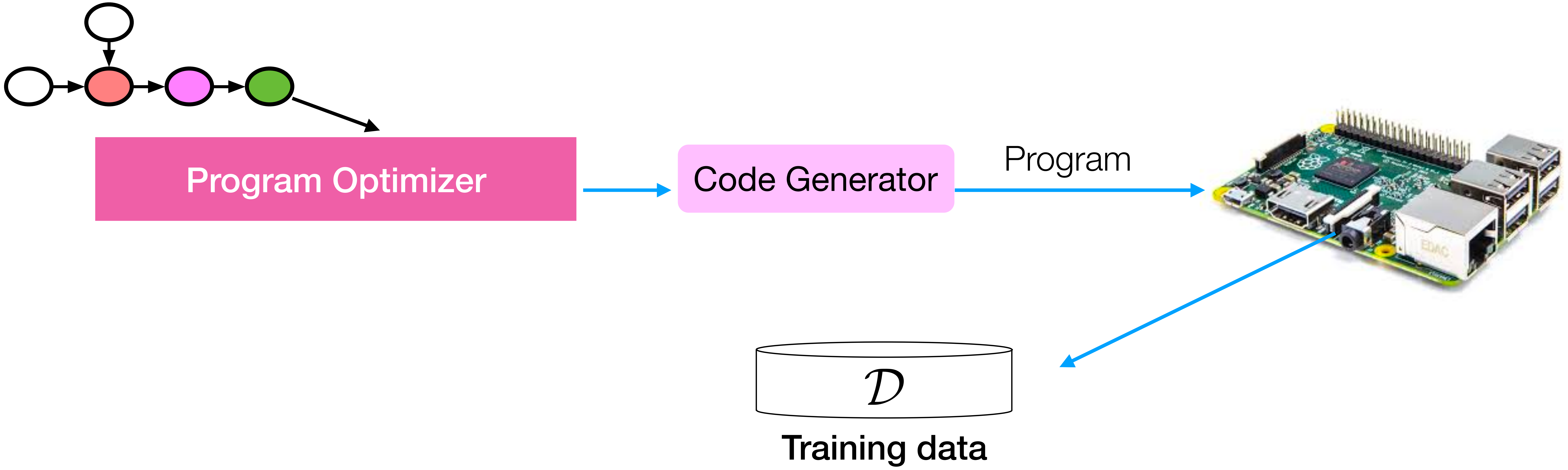
Latency Hiding

Hardware

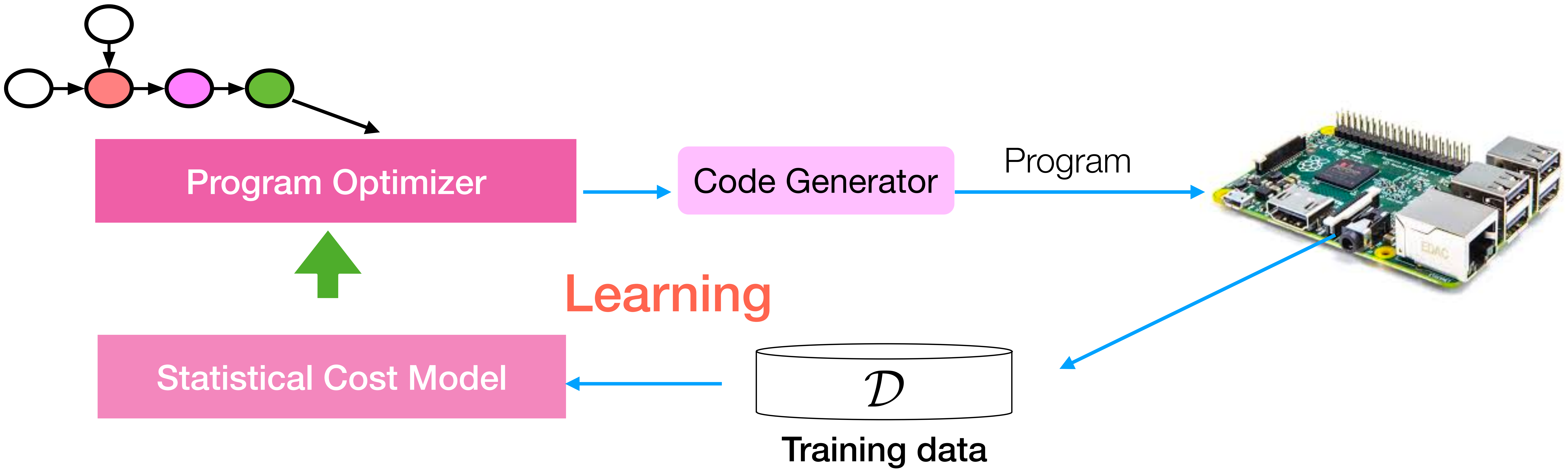
Learning-based Program Optimizer



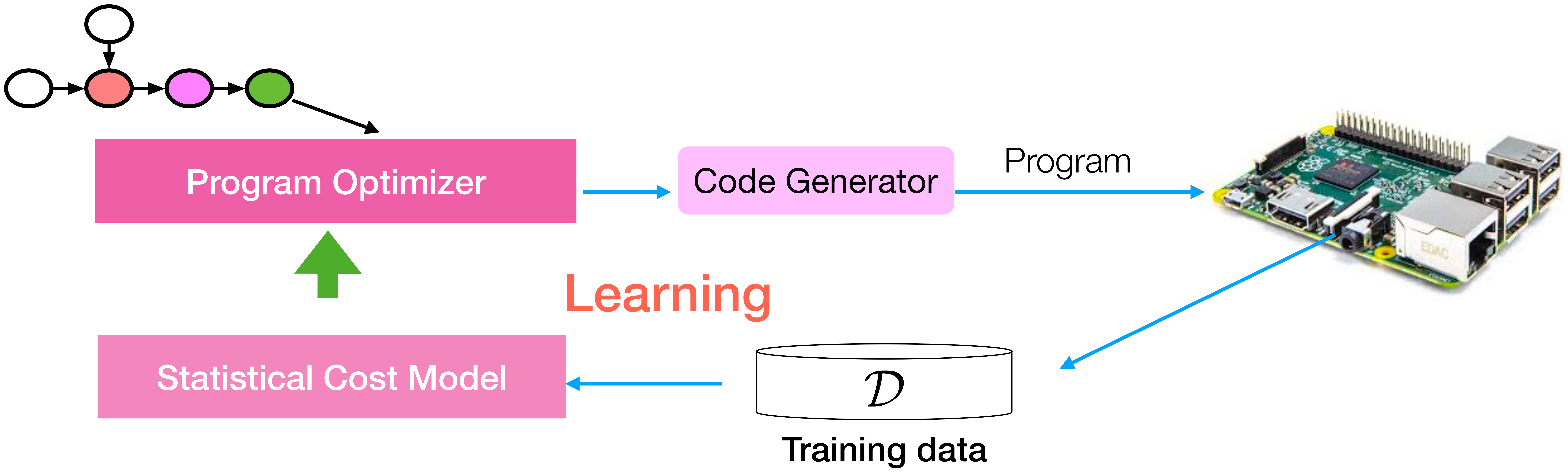
Learning-based Program Optimizer



Learning-based Program Optimizer



Learning-based Program Optimizer



Unique Problem Characteristics

- Relatively low experiment cost
- Domain-specific problem structure
- Large quantity of similar tasks

Program-aware Cost Modeling

High-Level Configuration

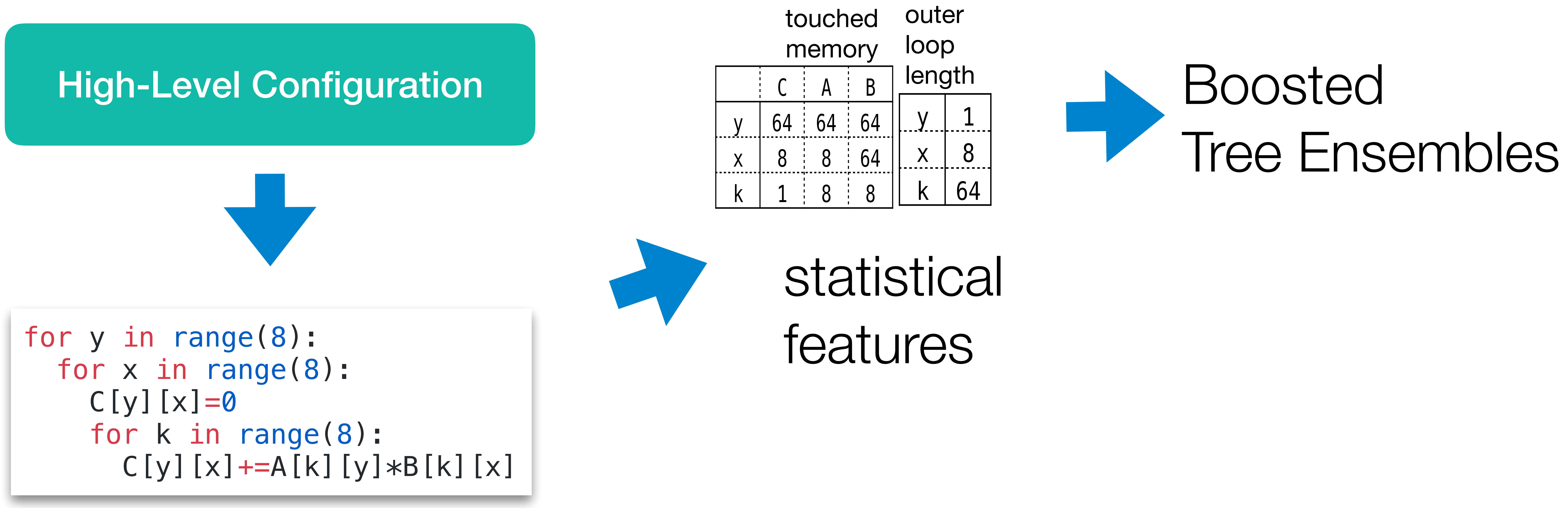
Program-aware Cost Modeling

High-Level Configuration

```
for y in range(8):
  for x in range(8):
    C[y][x]=0
    for k in range(8):
      C[y][x]+=A[k][y]*B[k][x]
```

Low-level Abstract Syntax Tree
(shared between tasks)

Program-aware Cost Modeling



```
for y in range(8):
  for x in range(8):
    C[y][x]=0
    for k in range(8):
      C[y][x]+=A[k][y]*B[k][x]
```

touched memory

	C	A	B
y	64	64	64
x	8	8	64
k	1	8	8

outer loop length

y	1
x	8
k	64

Low-level Abstract Syntax Tree
(shared between tasks)

Program-aware Cost Modeling

High-Level Configuration

```
for y in range(8):
  for x in range(8):
    C[y][x]=0
    for k in range(8):
      C[y][x]+=A[k][y]*B[k][x]
```

Low-level Abstract Syntax Tree
(shared between tasks)

touched memory

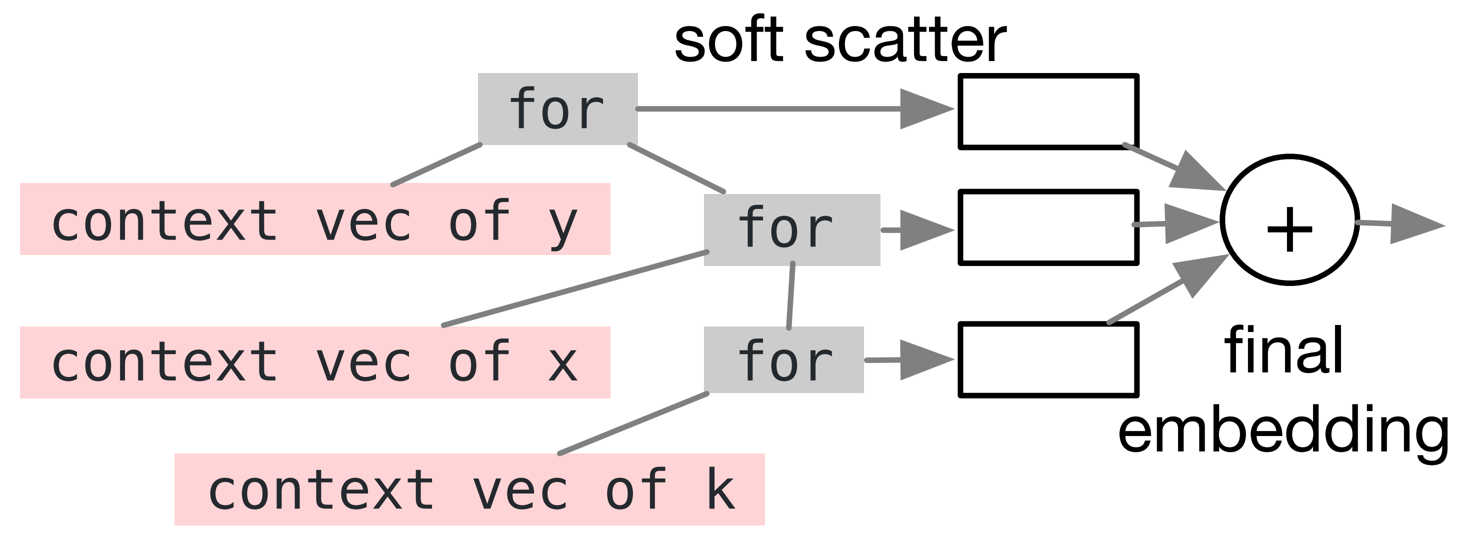
	C	A	B
y	64	64	64
x	8	8	64
k	1	8	8

outer loop length

y	1
x	8
k	64

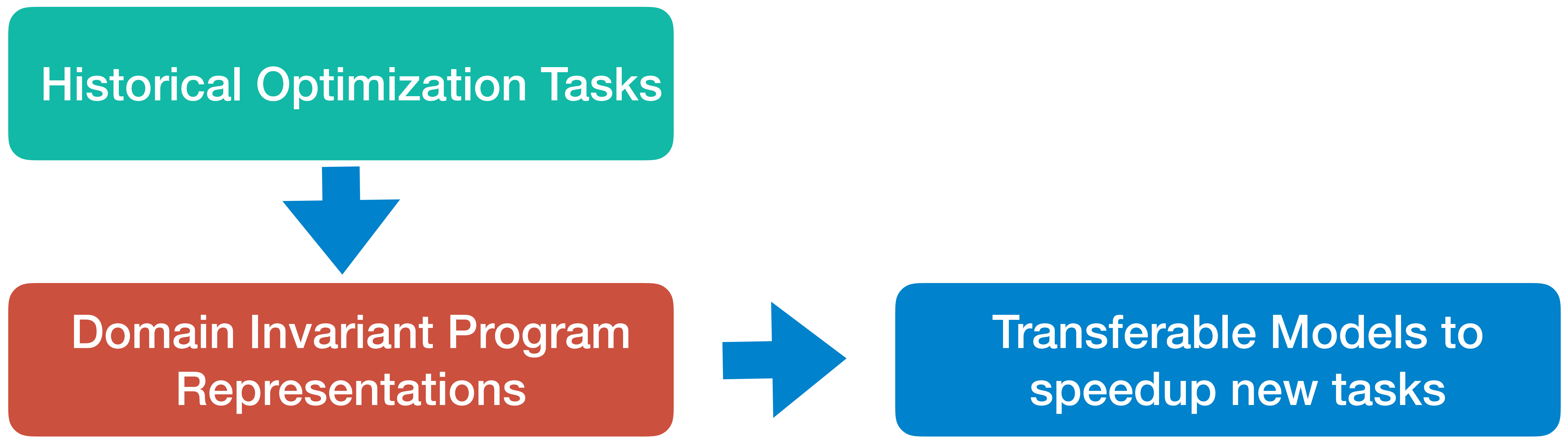
statistical features

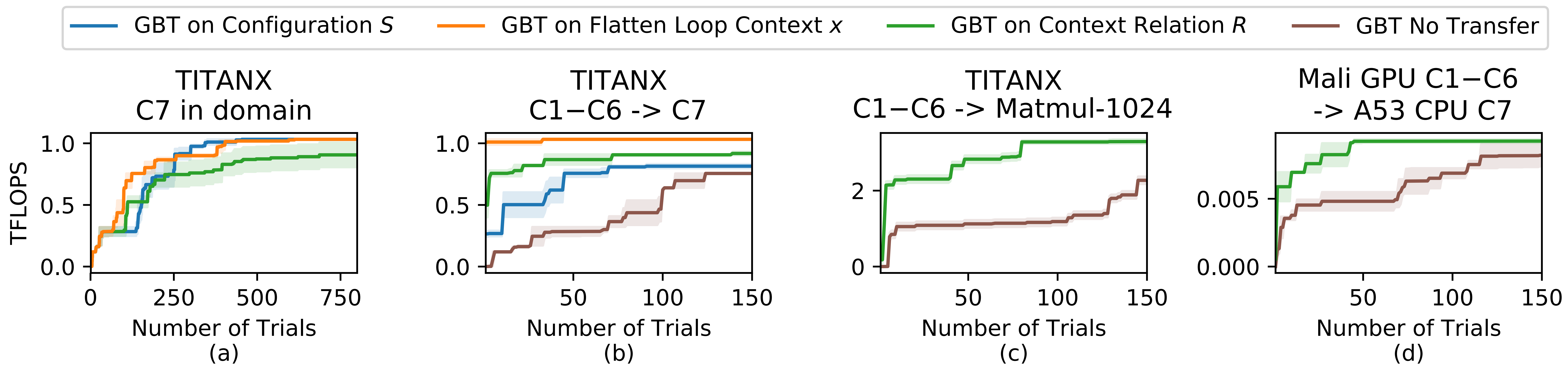
Boosted Tree Ensembles



TreeGRU

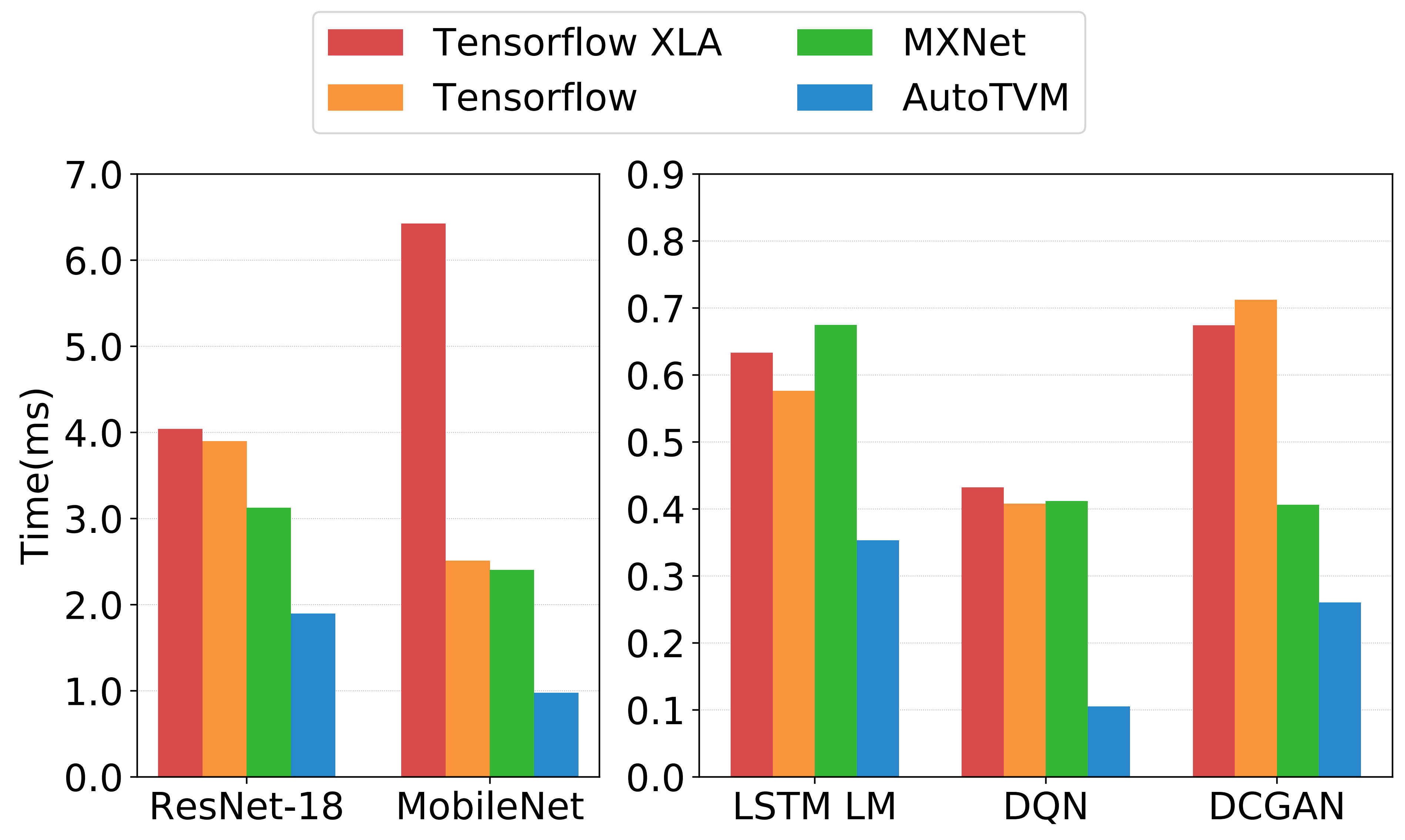
Transfer Learning Among Different Workloads


Historical Optimization Tasks
Domain Invariant Program Representations
Transferable Models to speedup new tasks

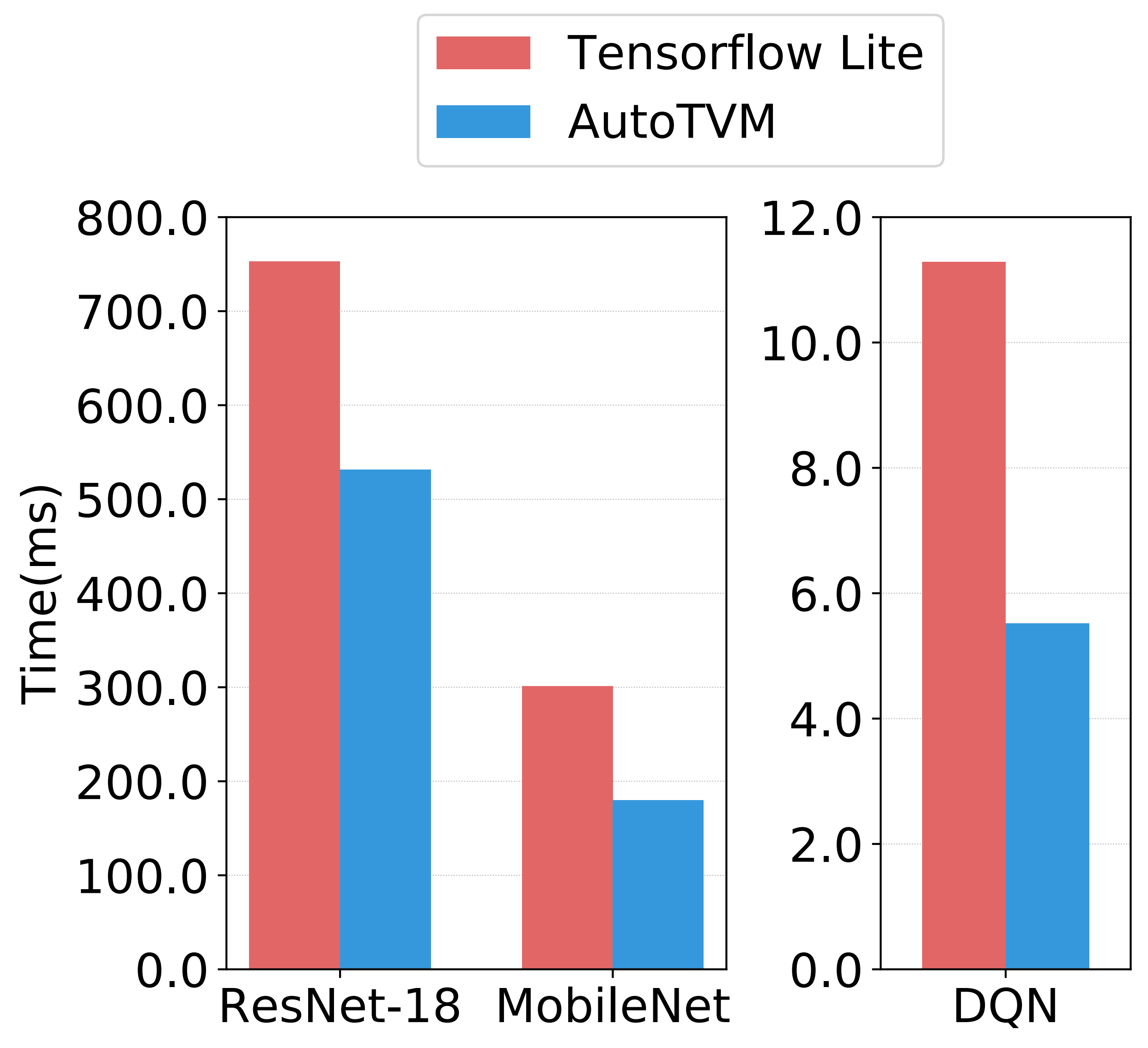


State of Art Performance

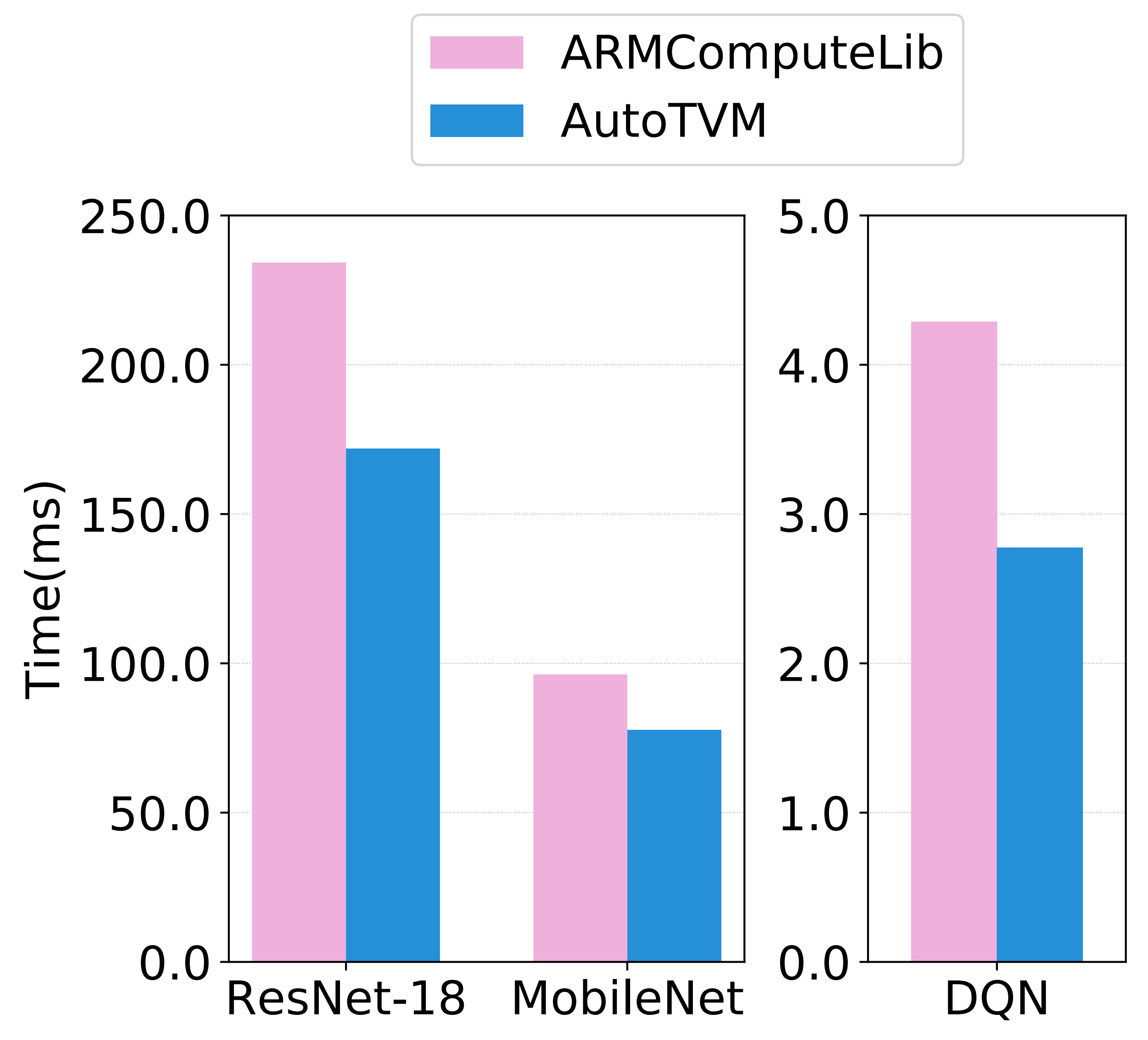
Nvidia GPU



ARM CPU



ARM GPU

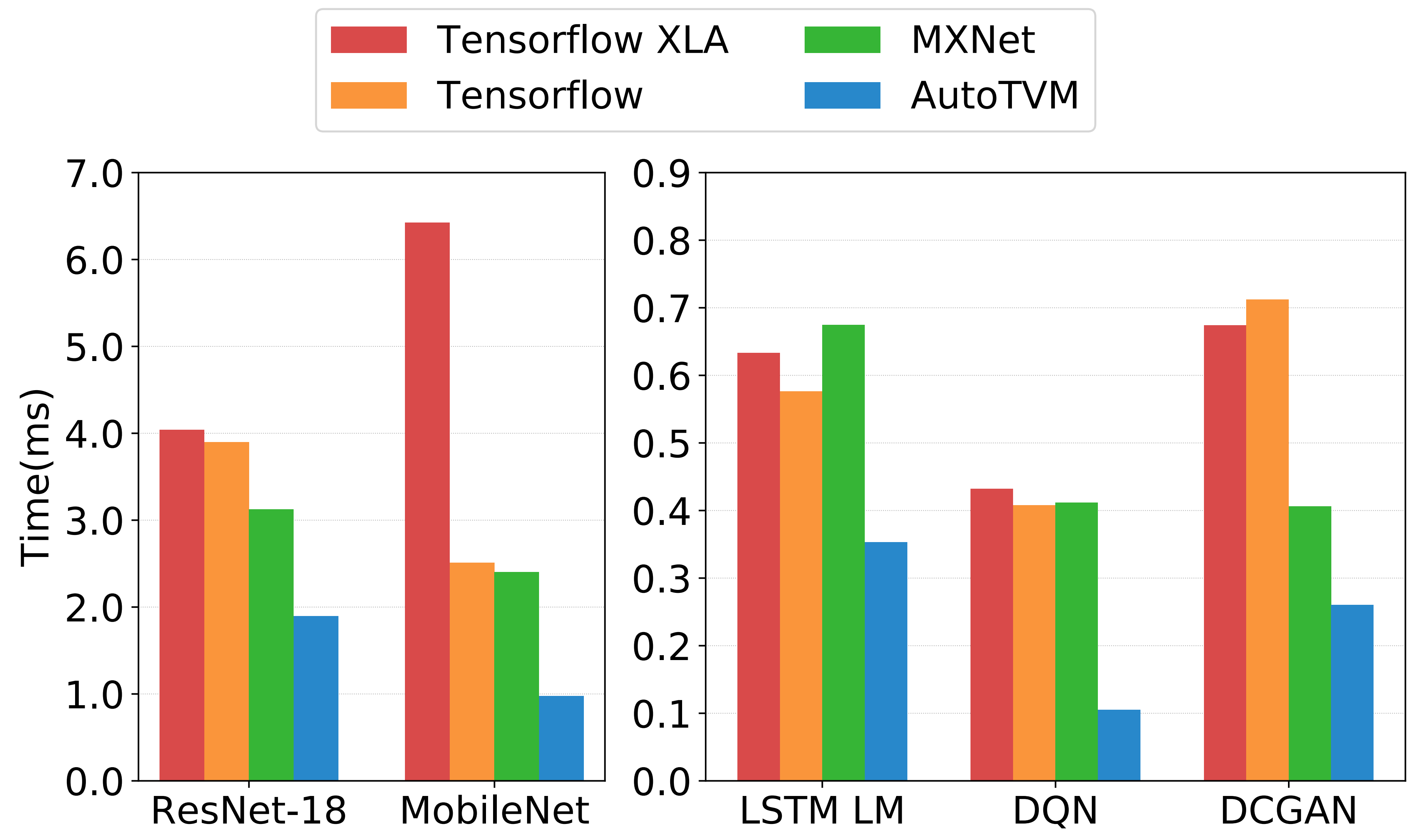


tvm.ai

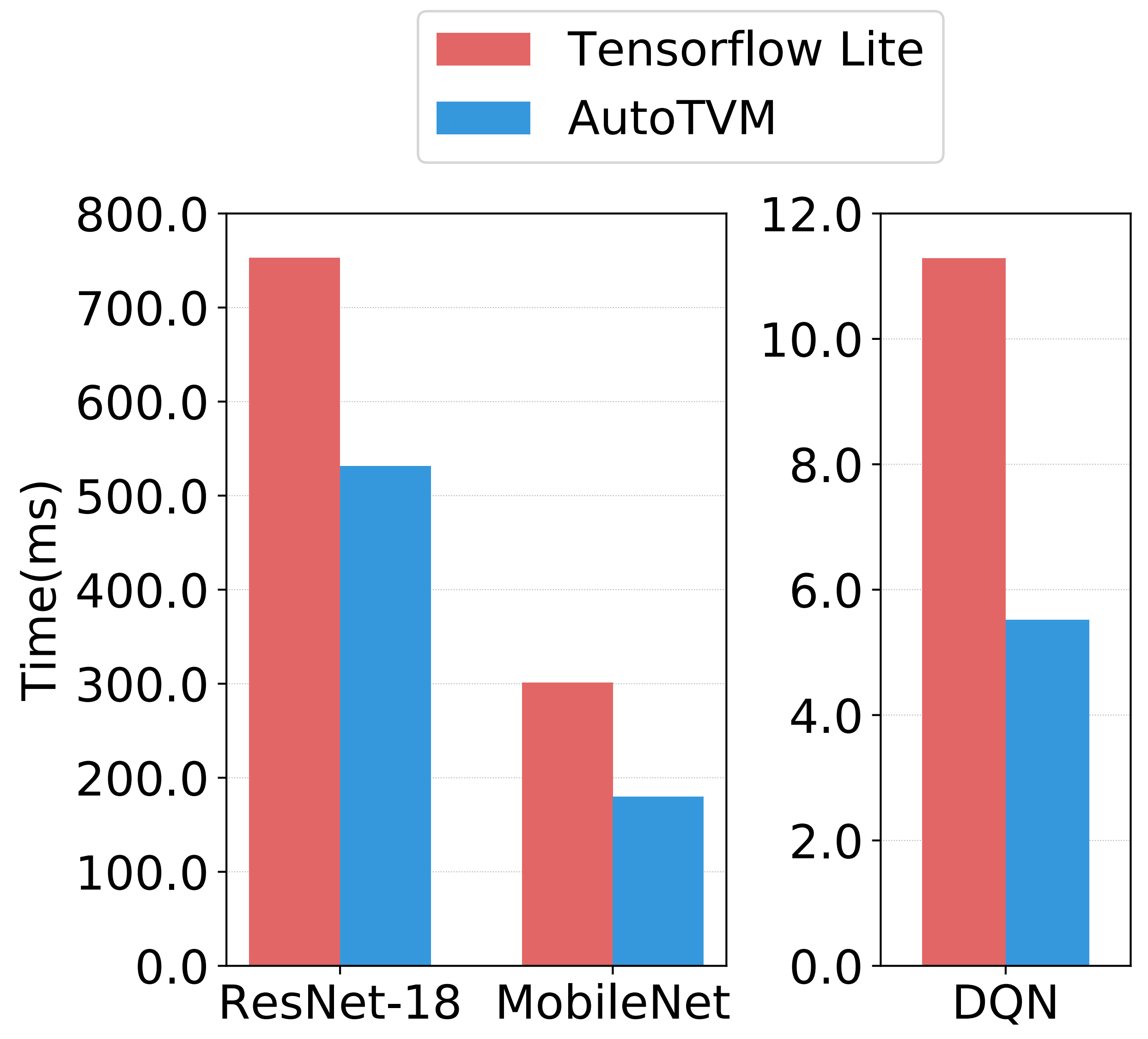
Poster #104

State of Art Performance

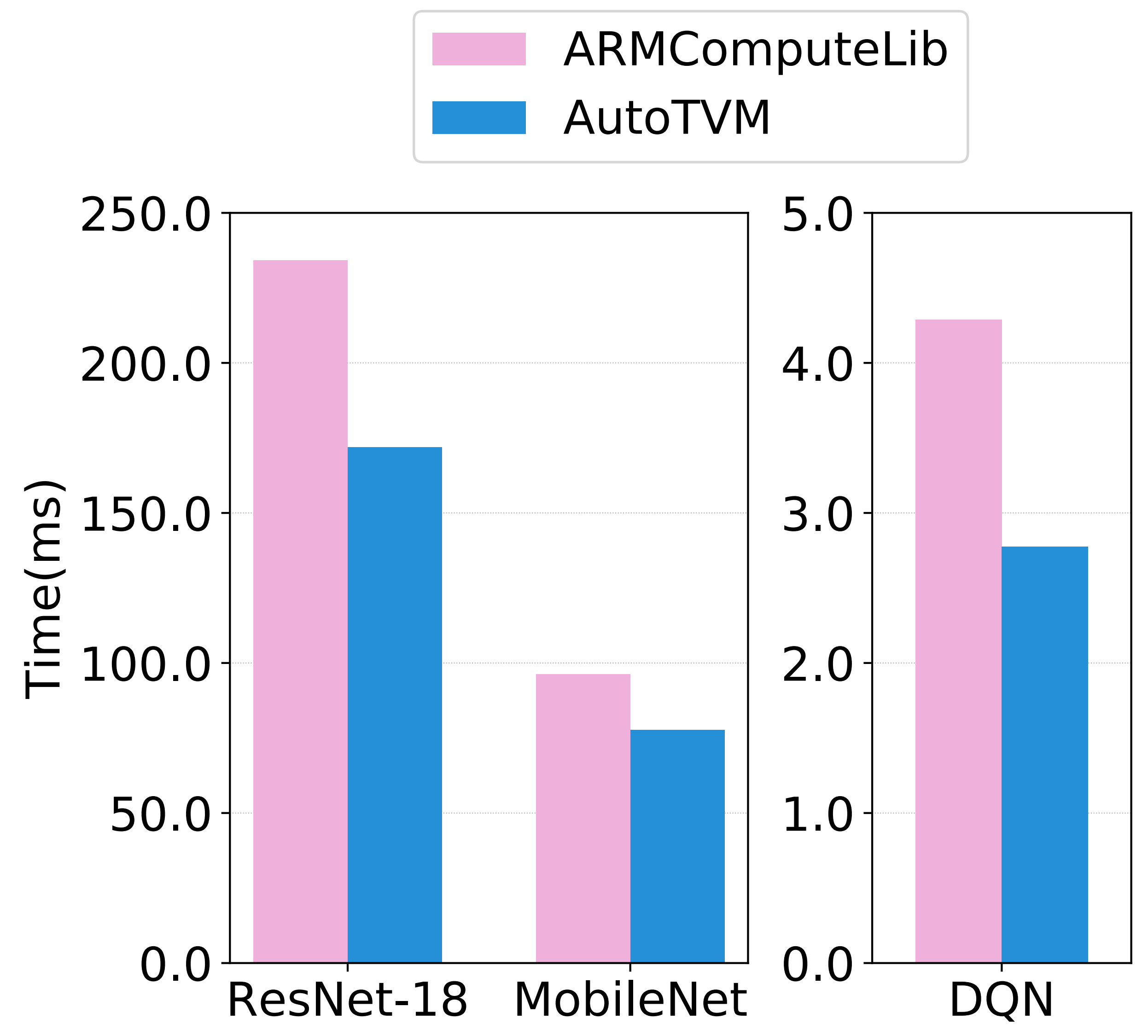
Nvidia GPU



ARM CPU



ARM GPU



In production use inside several major companies

tvm.ai

Poster #104