

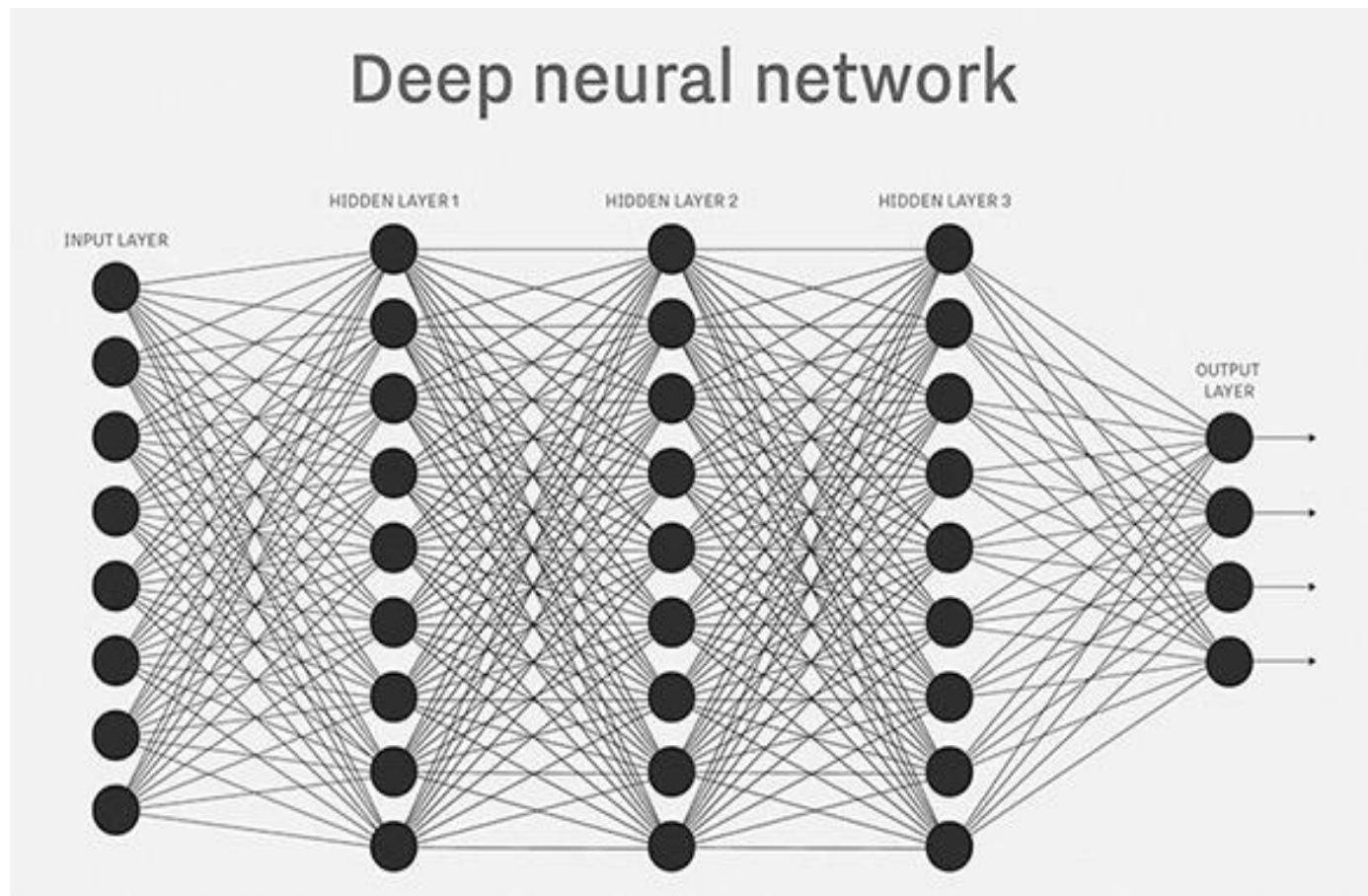


# Vitis AI

Olivier Méhaignerie  
Xilinx Field Application Engineer  
Avnet-Silica



# Deep Learning



5-Layer Neural Network

## ➤ Why now?

- New processors making DNN training feasible (Ops/\$)
- Huge amounts of training data

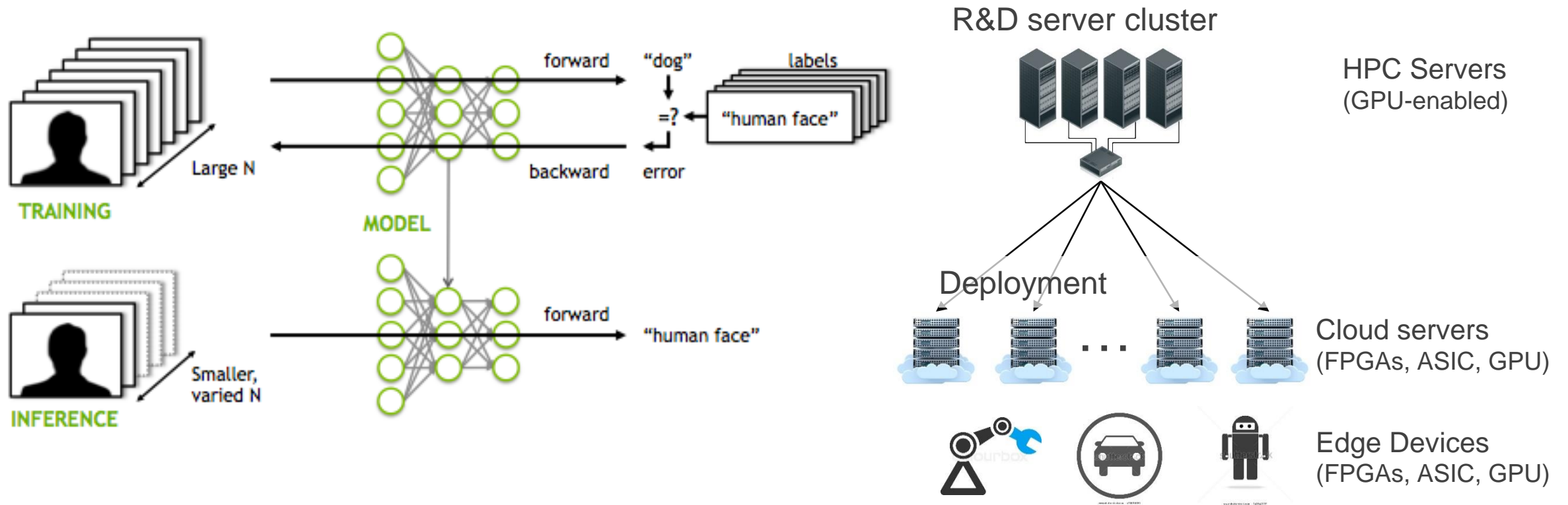
## ➤ Faster/Better Results

- Caused explosion in AI Research
- More Applications
- More Startups
- More Innovation
- More Acquisitions . . .

# Deep Learning: Training vs. Inference

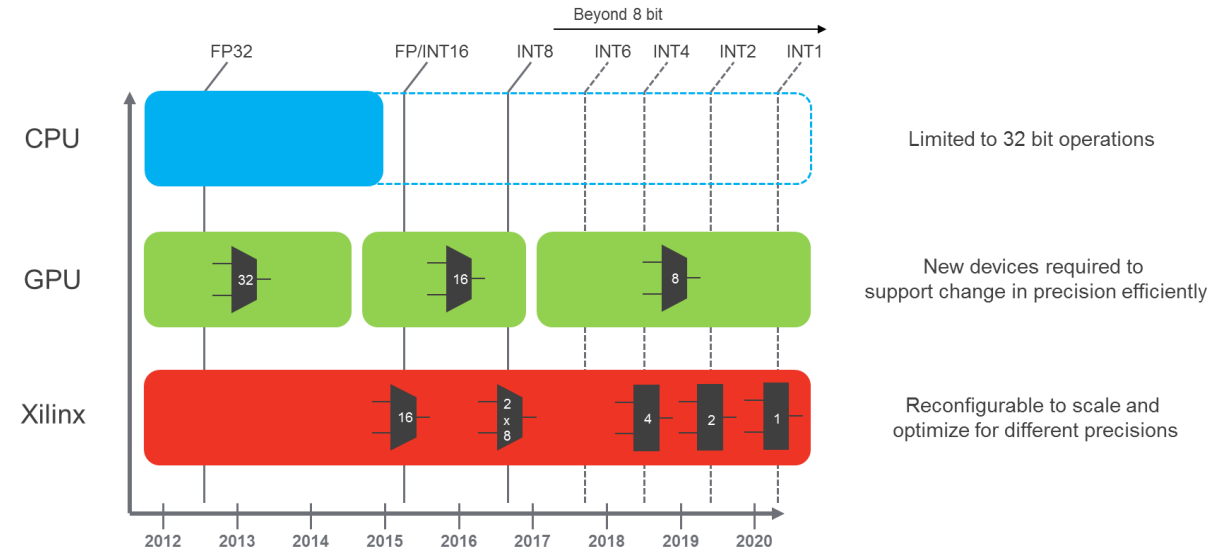
**Training:** Process for machine to “learn” and optimize a model from data

**Inference:** Using trained model to predict/estimate outcomes from new observations

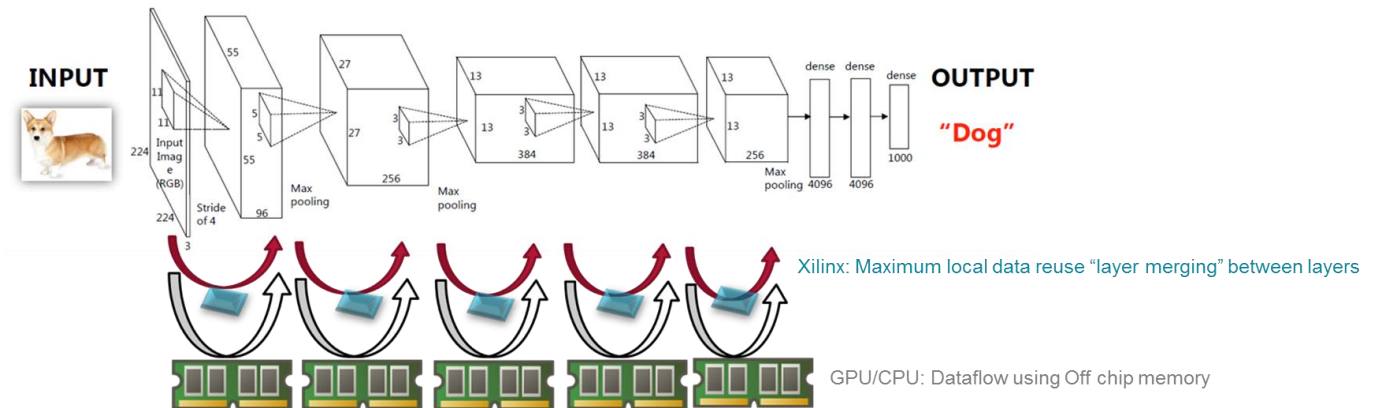


# Xilinx Features for Implementing Efficient Inference Engines

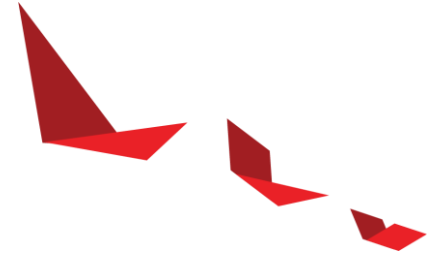
**Flexible Architecture for Any Precision**



**Flexible On-chip Memory for low latency**

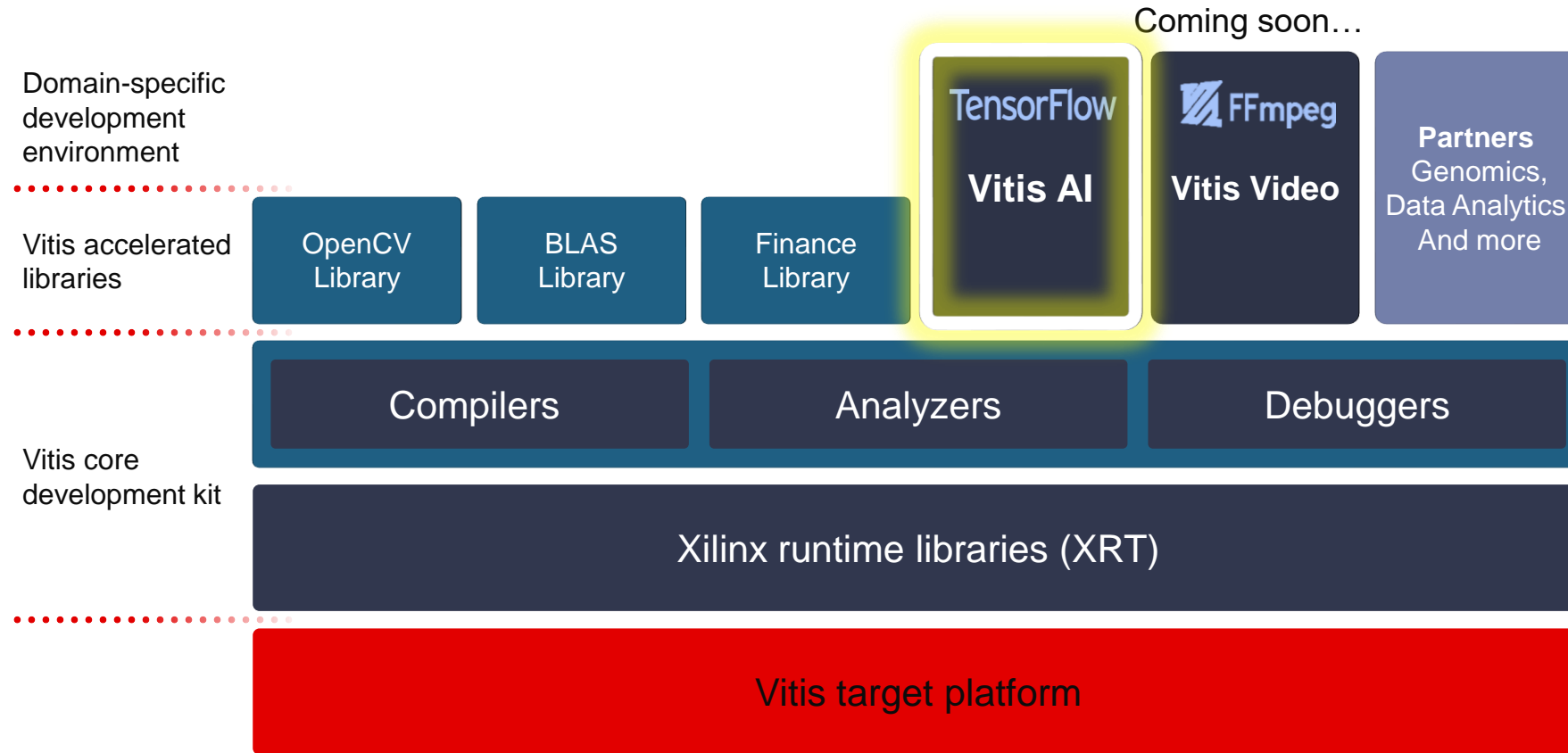


# AI Solutions for Xilinx



- ▶ Xilinx's Vitis AI (this presentation)
    - Evolution of DNNDK (acquisition of Deephi company)
    - Zynq, Alveo, Versal (soon)
  
  - ▶ FINN - <https://xilinx.github.io/finn/>
    - Launched by Xilinx Research Labs
    - Official support is community-based
    - Produces an AXI-attached microarchitecture
  
  - ▶ HLS4ML - <https://github.com/hls-fpga-machine-learning/hls4ml>
    - Open-source project
    - Generates a RTL microarchitecture
    - Leverages Vivado HLS for RTL generation
  
  - ▶ Mipsology's Zebra – [www.mipsology.com](http://www.mipsology.com)
    - Xilinx partner
- 5 - The easiest and most efficient solution

# Vitis AI: Unified AI Inference Solution Stack



# Vitis AI: Unified AI Inference Solution Stack

User Applications and Demo Zoo



Frameworks



Caffe

PyTorch

Vitis AI Models

Model Zoo

Custom Models

Vitis AI Development Kit

AI Optimizer

AI Quantizer

AI Compiler

AI Profiler

AI Library

- > Support both edge and cloud
- > Support AI model zoo
- > Several releases before this Vitis AI release

Vitis Runtime

Deep Learning Processing Unit (DPU)

DPU

CNN-Zynq

CNN-Alveo


LSTM-Alveo

CNN-AIE

LSTM-AIE


...

# Model Zoo




The diagram features a central black circle with the text "Vitis AI Model Zoo" in red. This central circle is surrounded by a network of red lines connecting eight smaller circular icons. The icons represent various AI concepts: a brain with a gear, a computer monitor with a brain, a head with a brain, a smartphone with a brain, a gear with a brain, a brain with a gear, a brain with a gear, and a brain with a gear.


**Rich models from Tensorflow and Caffe**




**Open and Free for all developers**



**Advanced optimization, including pruning, applied**



**Retrainable with custom dataset**





# Vitis AI Model Zoo 1.0

- ▶ Shared Repository of Pre-Trained AI Models
  - Ready to Deploy, Pre-Optimized Models
  - 50+ Models Support Broad Range of Applications
  - Open and Available on [GitHub](#)
- ▶ Leverage Standard Frameworks, Networks, Datasets
  - Trained Using TensorFlow and Caffe
- ▶ Deploy As-is, Re-Train or Further Optimize
  - Caffe\_Xilinx, a custom distribution of Caffe provided to test & finetune caffe models
  - Training code, test code and train eval instructions provided

Application	Model
Face	Face detection
	Landmark Localization
	Face recognition
	Face attributes recognition
Pedestrian	Pedestrian Detection
	Pose Estimation
	Person Re-identification
Video Analytics	Object detection
	Pedestrian Attributes Recognition
	Car Attributes Recognition
	Car Logo Detection
	Car Logo Recognition
	License Plate Detection
	License Plate Recognition
ADAS/AD	Object Detection
	3D Car Detection
	Lane Detection
	Traffic Sign Detection
	Semantic Segmentation
	Drivable Space Detection

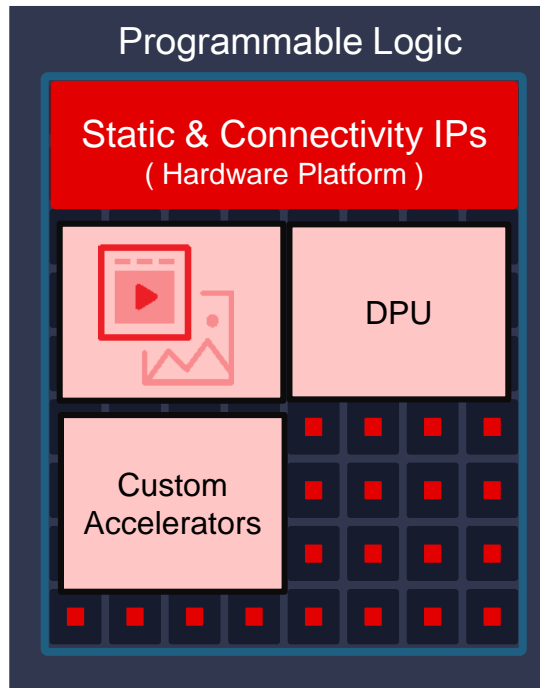
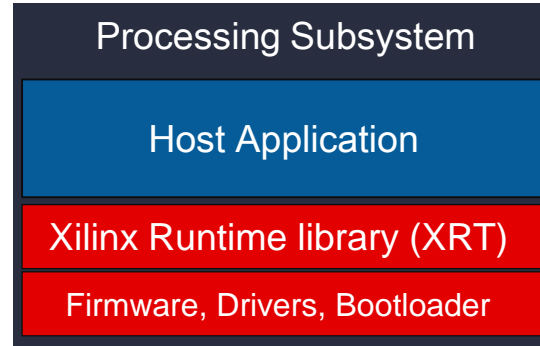
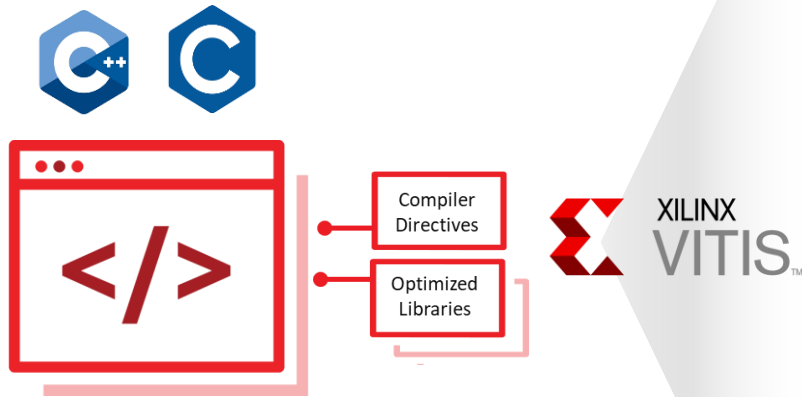
# Model Zoo

Name	Framework	Backbone	Input Size	OPS per image	Traini
cf_resnet50_imagenet_224_224_7.7G	caffe	resnet50	224*224	7.7G	ImageNet
cf_resnet18_imagenet_224_224_3.65G	caffe	resnet18	224*224	3.65G	ImageNet
cf_inceptionv1_imagenet_224_224_3.16G	caffe	inception_v1	224*224	3.16G	ImageNet
cf_inceptionv2_imagenet_224_224_4G	caffe	bn-inception	224*224	4G	ImageNet
cf_inceptionv3_imagenet_299_299_11.4G	caffe	inception_v3	299*299	11.4G	ImageNet
cf_inceptionv4_imagenet_299_299_24.5G	caffe	inception_v3	299*299	24.5G	ImageNet
cf_mobilenetv2_imagenet_224_224_0.59G	caffe	MobileNet_v2	224*224	608M	ImageNet

```
|— test_code                                # Contains code and instructions.
|   |— float                                # Test code and instruction for floating model for evaluation.
|   |   |— quantized                        # Test code and instruction for quantized model for evaluation.
|
|— readme.md                               # Contains the environment requirement, the input and output nodes as well as
|                                       the data preprocess and postprocess information.
|
|— quantized
|   |— deploy.model.pb                     # Quantized model for the compiler (extended Tensorflow format).
|   |— quantize_eval_model.pb              # Quantized model for evaluation.
|
|— float
|   |— frozen.pb                           # Float-point frozen model, the input to the `vai_q_tensorflow`.
```

# Vitis AI Or Vitis

## What's the Relation ?



```

100101010010
110010101011
001001010100
101100101010
110010010101
001011001010
DPU Instruction
    
```



## Vitis Target Platform

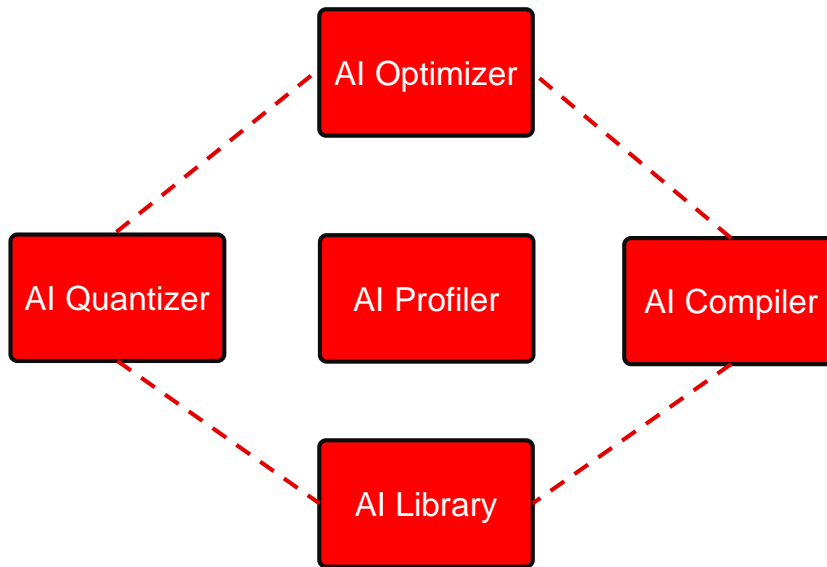
# Direct Framework Compilation In Minutes

PyTorch Caffe

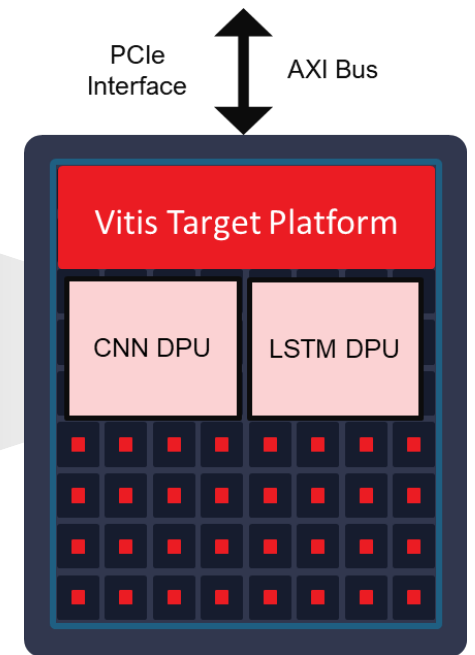
TensorFlow



Trained Models

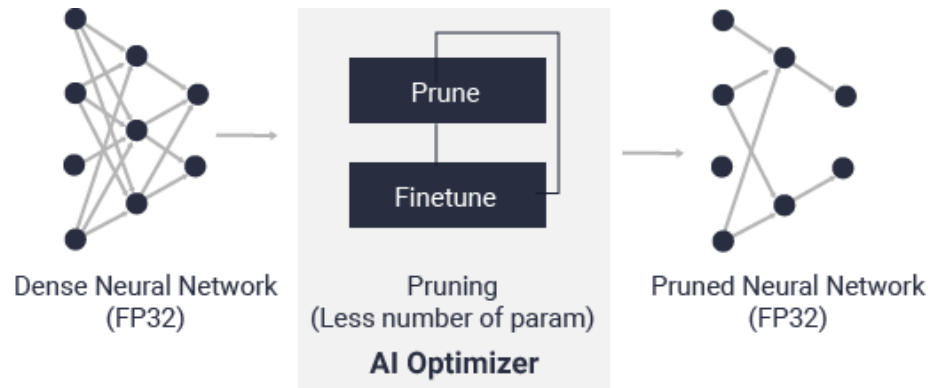


```
100101010010
110010101011
001001010100
101100101010
110010010101
001011001010
DPU Instruction
```



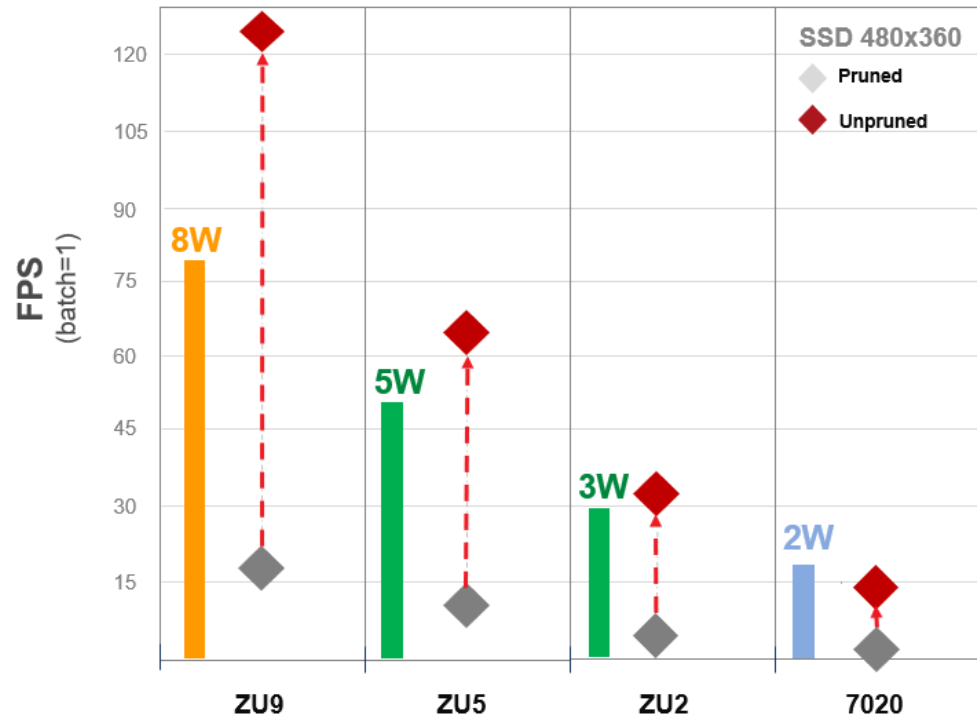
Vitis AI Development Kit

# AI Optimizer – pruning tool



## > World's leading model compression

- >> Iterative, coarse-grained pruning
- >> Reduce model size 5 – 30x
- >> Increase performance 2 – 10x
- >> Minimal accuracy loss, <1%



## > Supported framework

- >> Caffe, Darknet, TensorFlow

## > Commercial license available

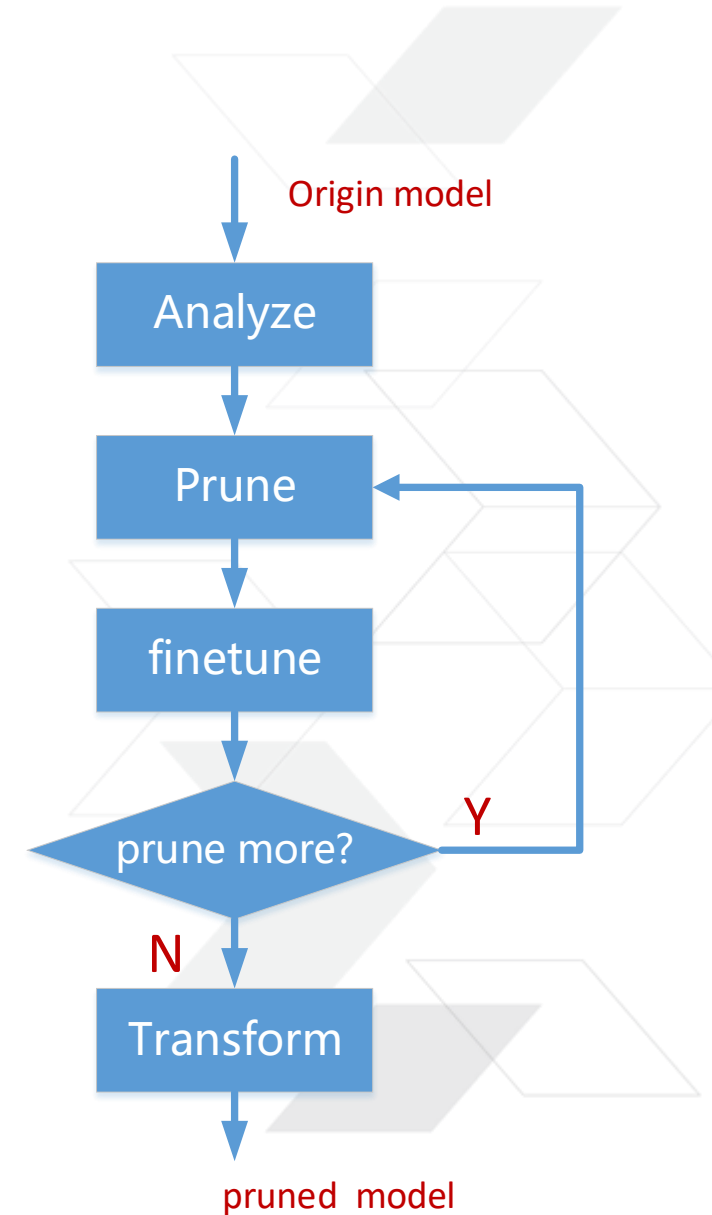
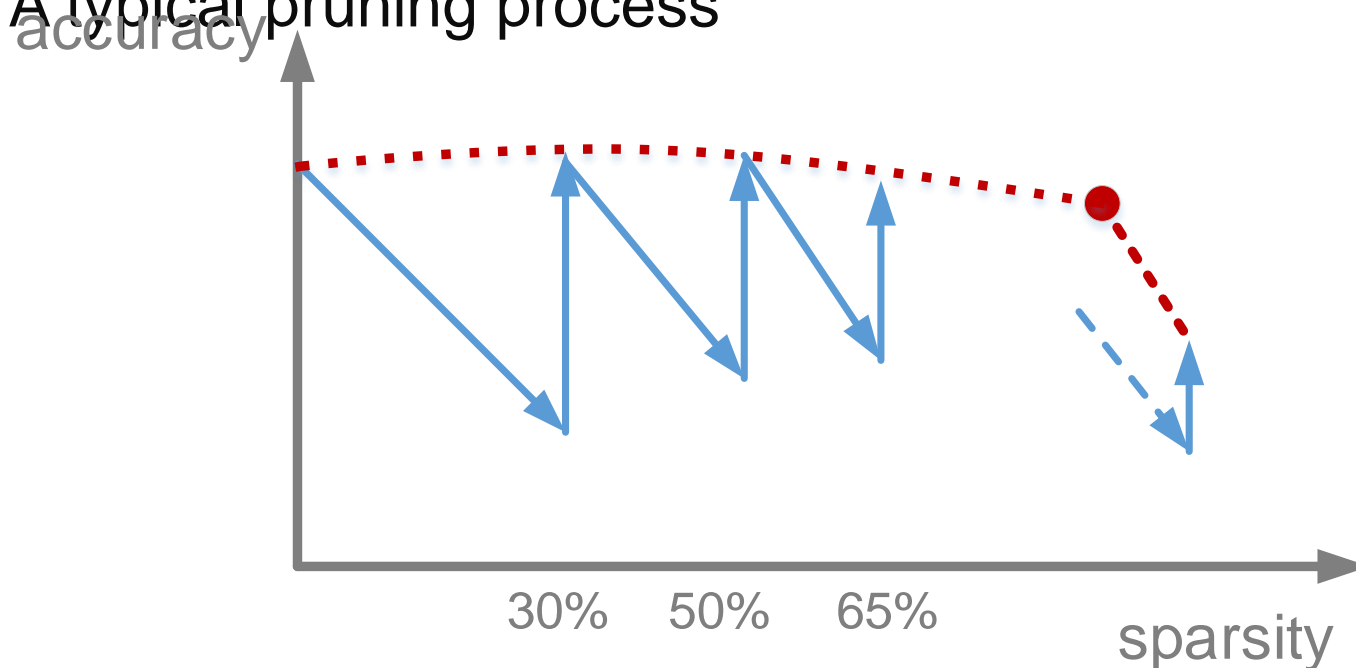
- >> Contact Xilinx sales representative

# Iterative Pruning

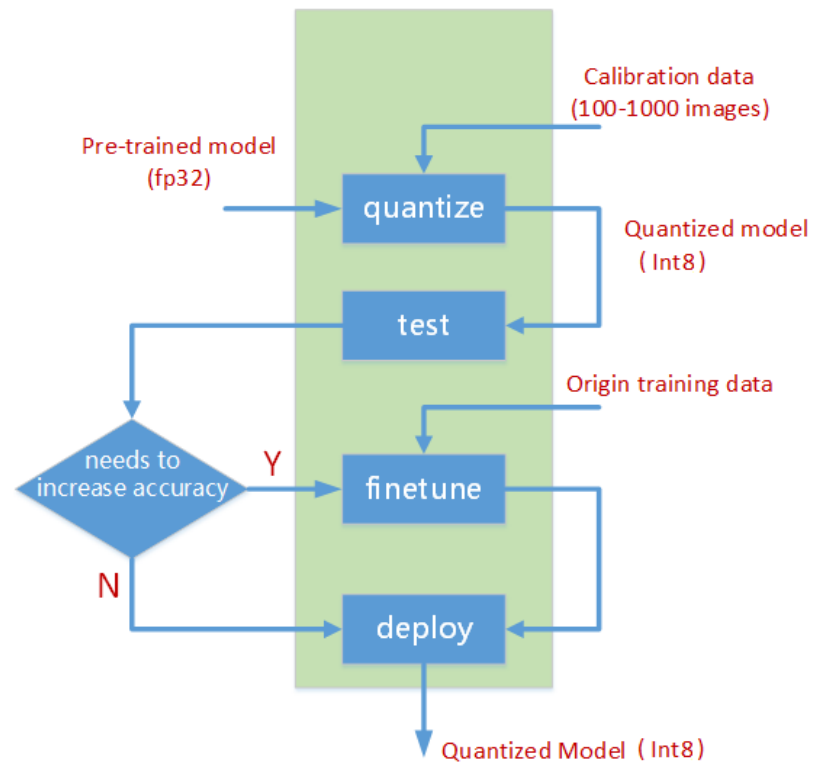
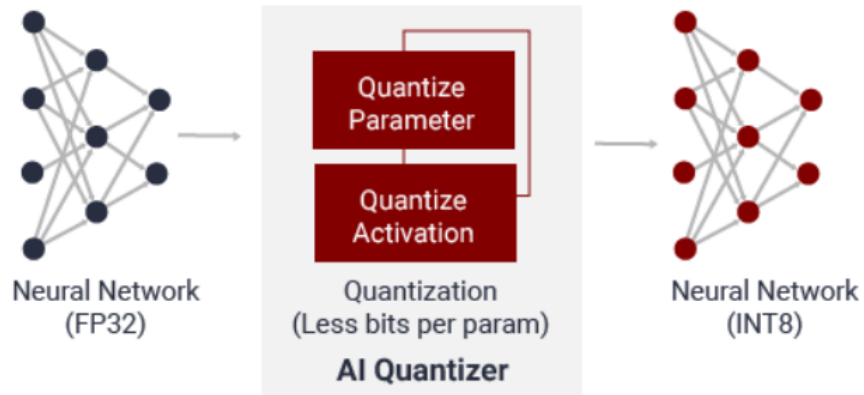
## ▶ Iterative pruning

- Iterative pruning generally works better than pruning directly
- When facing accuracy drops, try a small step
- For a full pruning work, set steps from large to small
- Experience on model training is very helpful

## ▶ A typical pruning process



# Vitis AI Quantizer



## > Uniform symmetric quantization

>> 8bit for both weights and activation

## > Support both calibration and finetune

>> Calibration – A small set of training data

>> Finetune – Original training data, further increase accuracy

## > Support framework

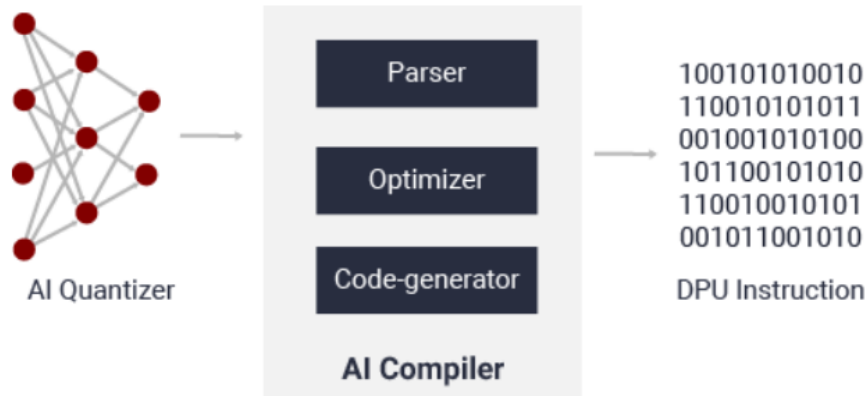
>> Caffe, Tensorflow

>> Pytorch

## > Have both GPU and CPU version

>> GPU version is 10x faster than CPU version

# AI Compiler



> **Maps the quantized model to instruction set and data flow**

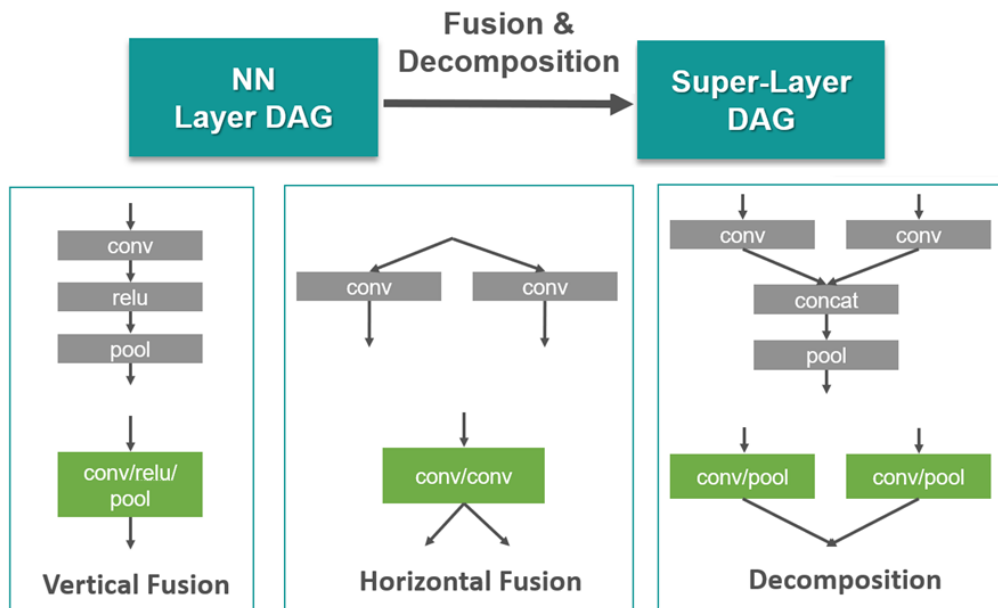
>> High-efficient tensor-level DPU instruction set

> **Performs sophisticated optimizations**

>> Layer fusion and decomposition

>> Instruction scheduling

>> Reuses on-chip memory as much as possible



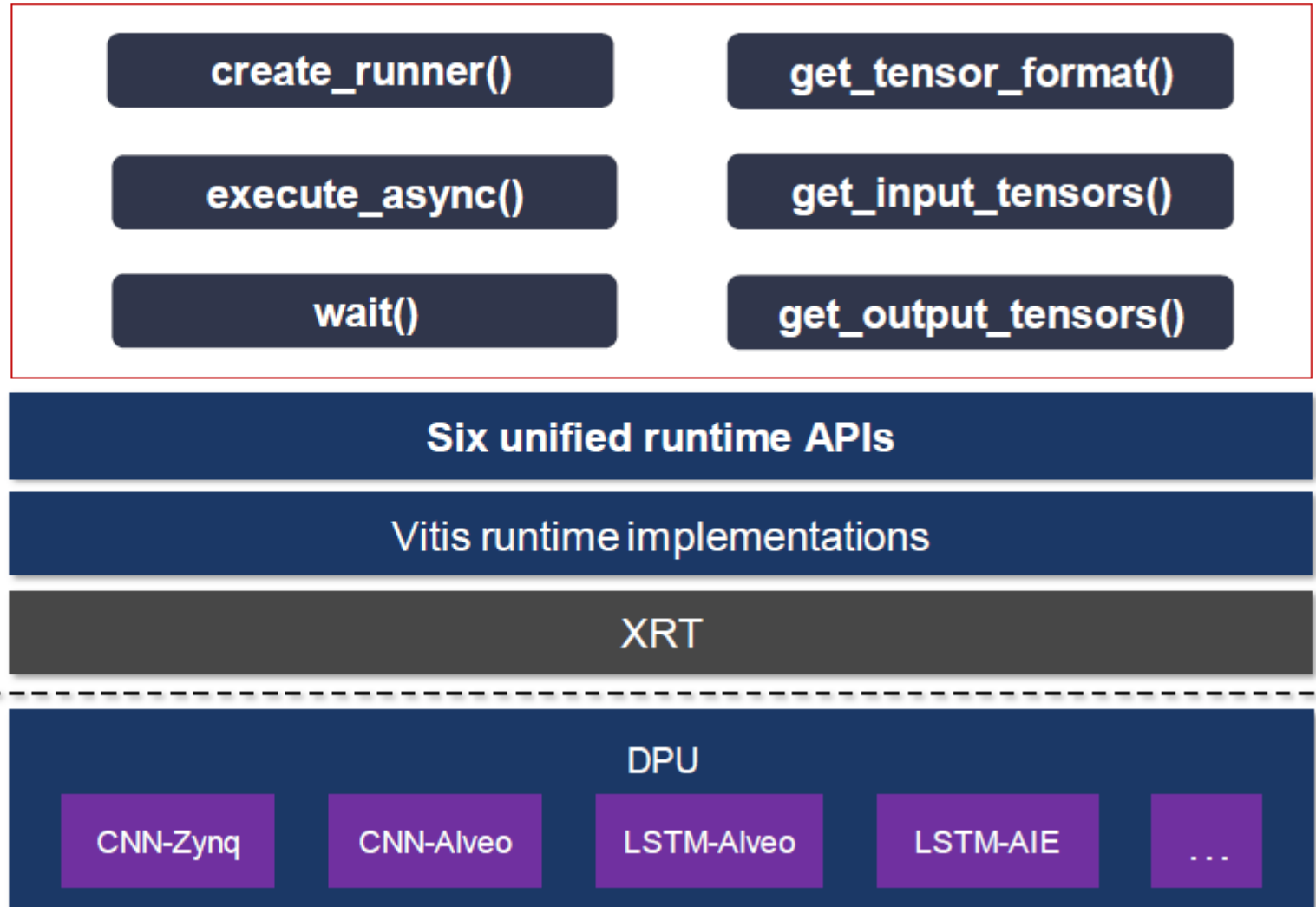
> **Support framework**

>> Caffe, Tensorflow

>> Pytorch (Q1, 2020)

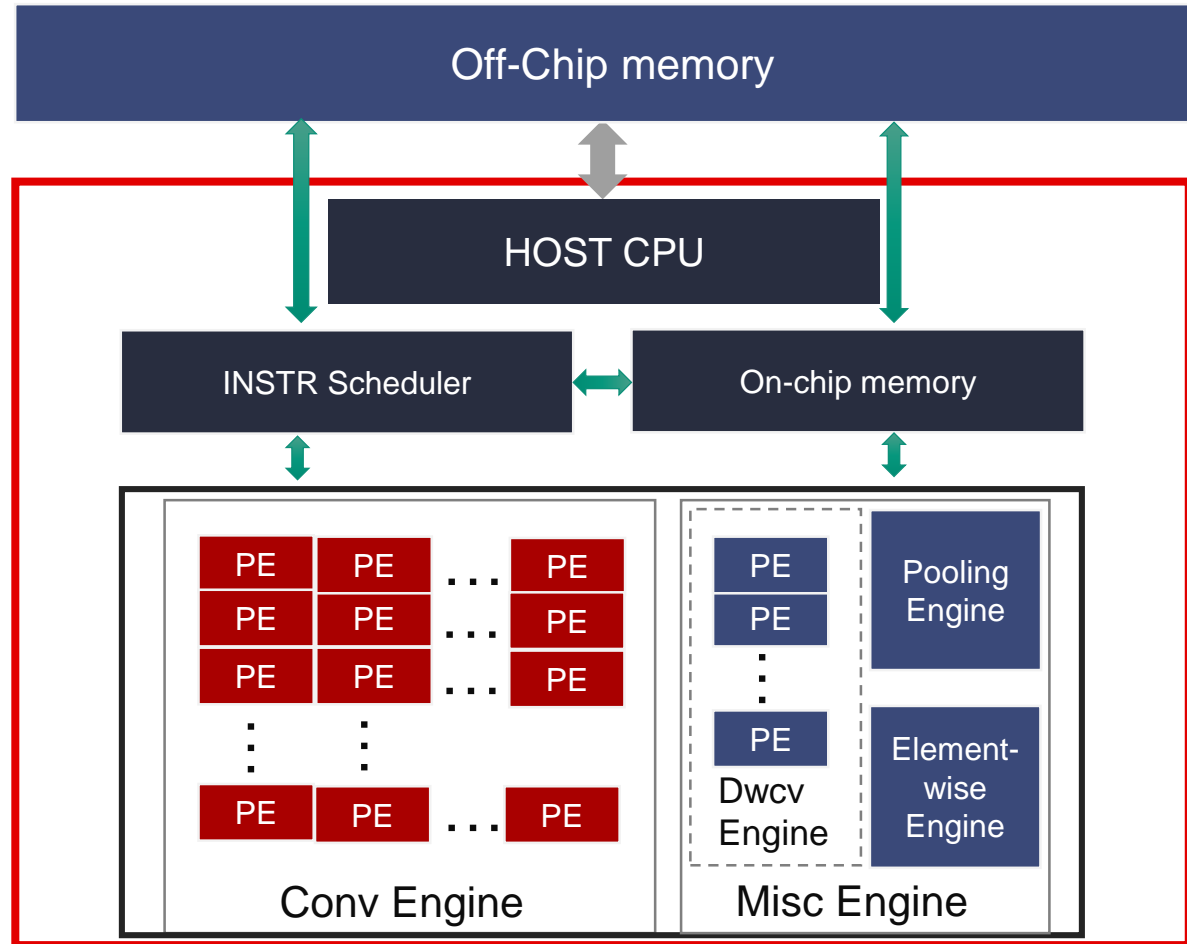


# Deploy Using Vitis AI runtime APIs



Unified Vitis AI runtime with same six APIs across edge and cloud

# CNN DPU for Zynq SoC / MPSoC



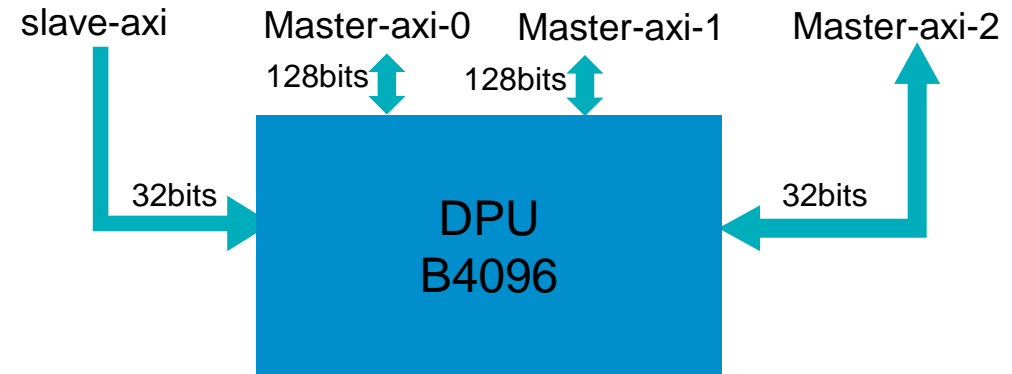
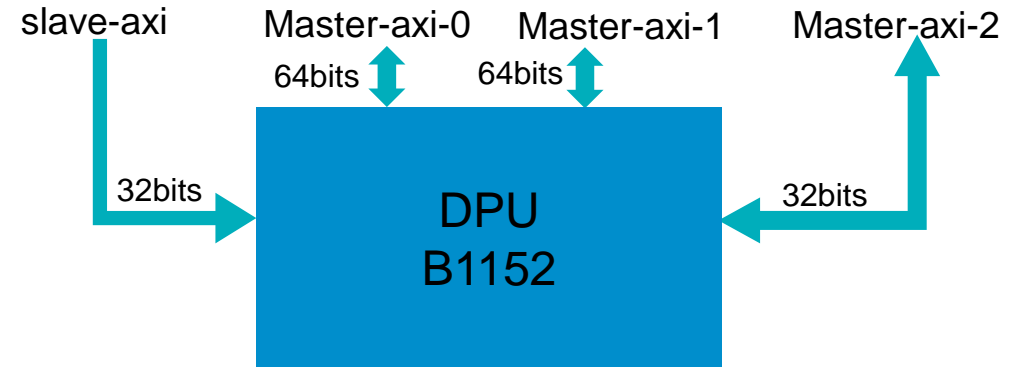
## > Flexible and Configurable DPU

- >> Configurable hardware architecture includes : Z7020 to Z7100, ZU2 to ZU11
- >> Relu, Relu6, LeakyRelu
- >> Max/Average pooling 2x2~8x8
- >> Ram usage for higher performance or lower resource utilization
- >> Core number, Bram or Uram, More DSP or less DSP
- >> Support channel augmentation to improve performance
- >> Support low power consumption feature

[https://www.xilinx.com/support/documentation/ip\\_documentation/dpu/v3\\_2/pg338-dpu.pdf](https://www.xilinx.com/support/documentation/ip_documentation/dpu/v3_2/pg338-dpu.pdf)

# DPU Typical Options & Interfaces

- ▶ 3-level parallelism is exploited
  - Pixel \* input channel \* output channel
- ▶ Small core - B1152
  - Parallelism: 4\*12\*12
  - target Z7020/ZU2/ZU3
- ▶ Big core - B4096
  - Parallelism: 8\*16\*16
  - Target ZU5 and above



# DPU Utilization

*More DSP*

Arch	LUTs	Registers	BRAM*	DSP
B512	17951	28280	69.5	97
B800	20617	35065	87	141
B1024	22327	39000	101.5	193
B1152	22796	40276	117.5	193
B1600	26270	50005	123	281
B2304	29592	57549	161.5	385
B3136	33266	69110	203.5	505
B4096	37495	84157	249.5	641

*More LUT*

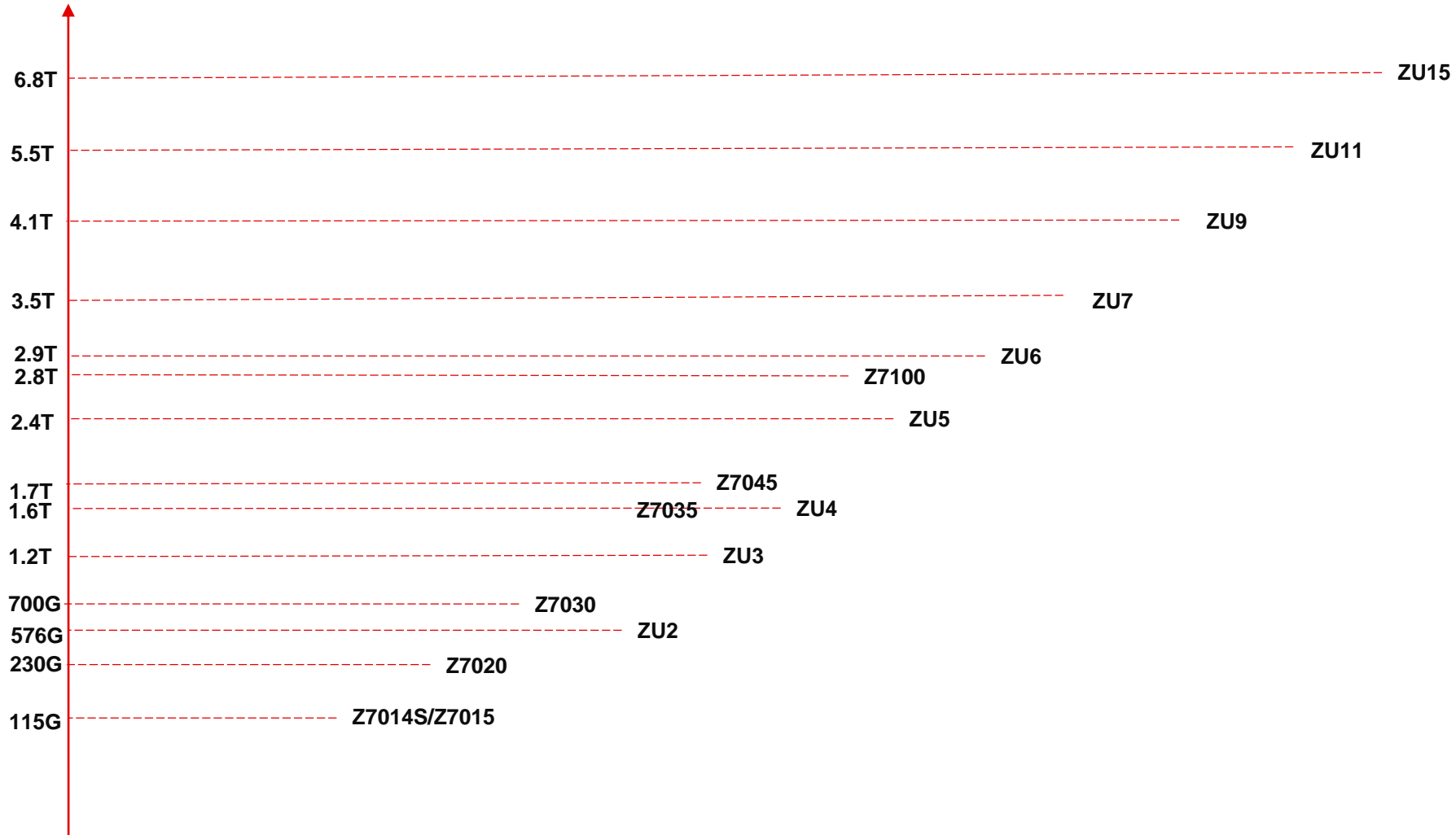
Arch	LUTs	Registers	BRAM*	DSP
B512	20759	33572	69.5	66
B1024	29155	49823	101.5	130
B1152	30043	49588	117.5	146
B1600	33130	60739	123	202
B2304	37055	72850	161.5	290
B3136	41714	86132	203.5	394
B4096	44583	99791	249.5	514

*DPU provides flexible option depending on customer's resources and continues to improve*

*\* URAM also can be used by DPU if device supports, every URAM is roughly used as 3.7 BRAM*

# DPU Scalability

Peak INT8 OPS\*



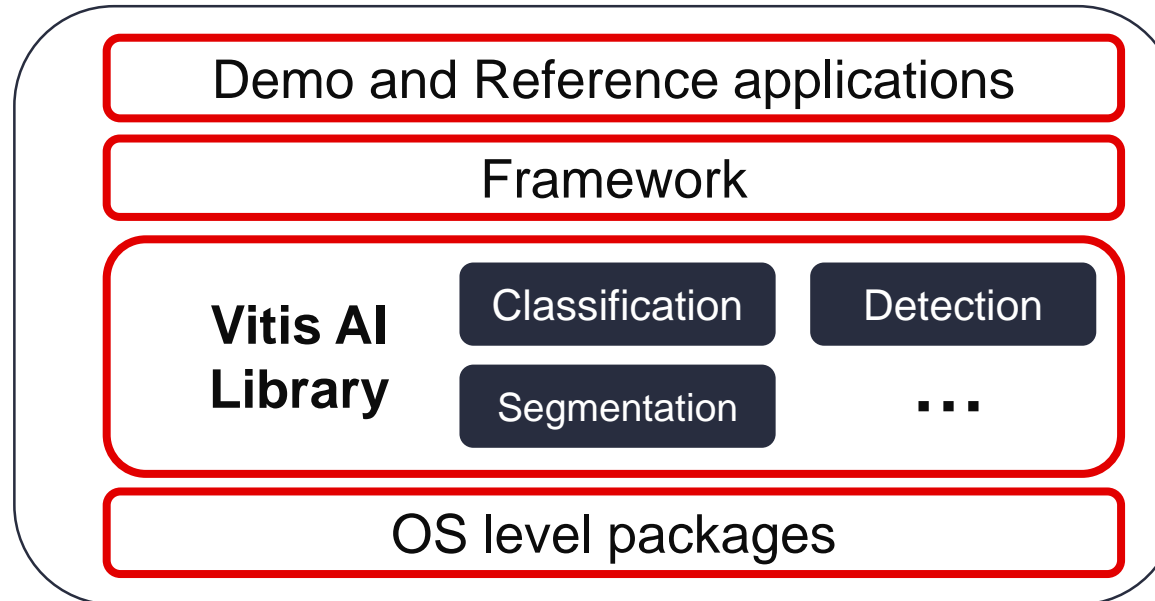
\* With heterogenous DPUs

# Vitis AI Library: the What?

- ▶ **Vitis AI Library** provides high-level API based libraries across different vision tasks: classification, detection, segmentation and etc.
  - Reference applications to help customers' fast prototyping
  - Optimized codes used in AI applications and products



User Applications



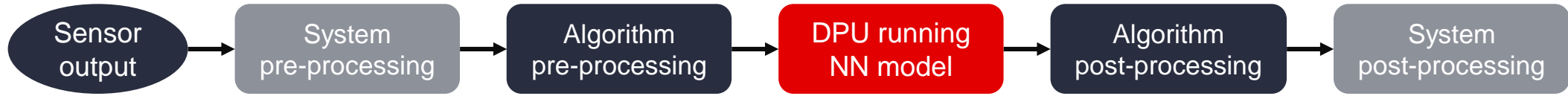
Ease-of-Use

Optimized

Open

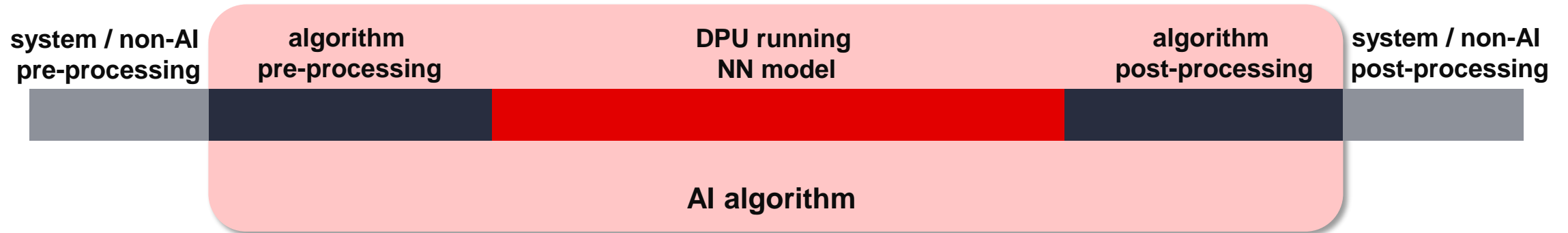
# AI Application General Processing Flow

- ▶ A typical abstraction of processing flow:



- › **Algorithm-level processing**
  - » Data normalization before sending to DPU
  - » Post processing (e.g. bounding boxes decoding in detection)
- › **Additional system-level workloads for AI inference**
  - » Color conversion / resizing
  - » Path planning / control / status update

# What Vitis AI Library Provides



- ▶ **AI Library offers libraries for**
  - Algorithm-level optimization
  - Open and easy to extend
  - Directly support models in AI Model Zoo



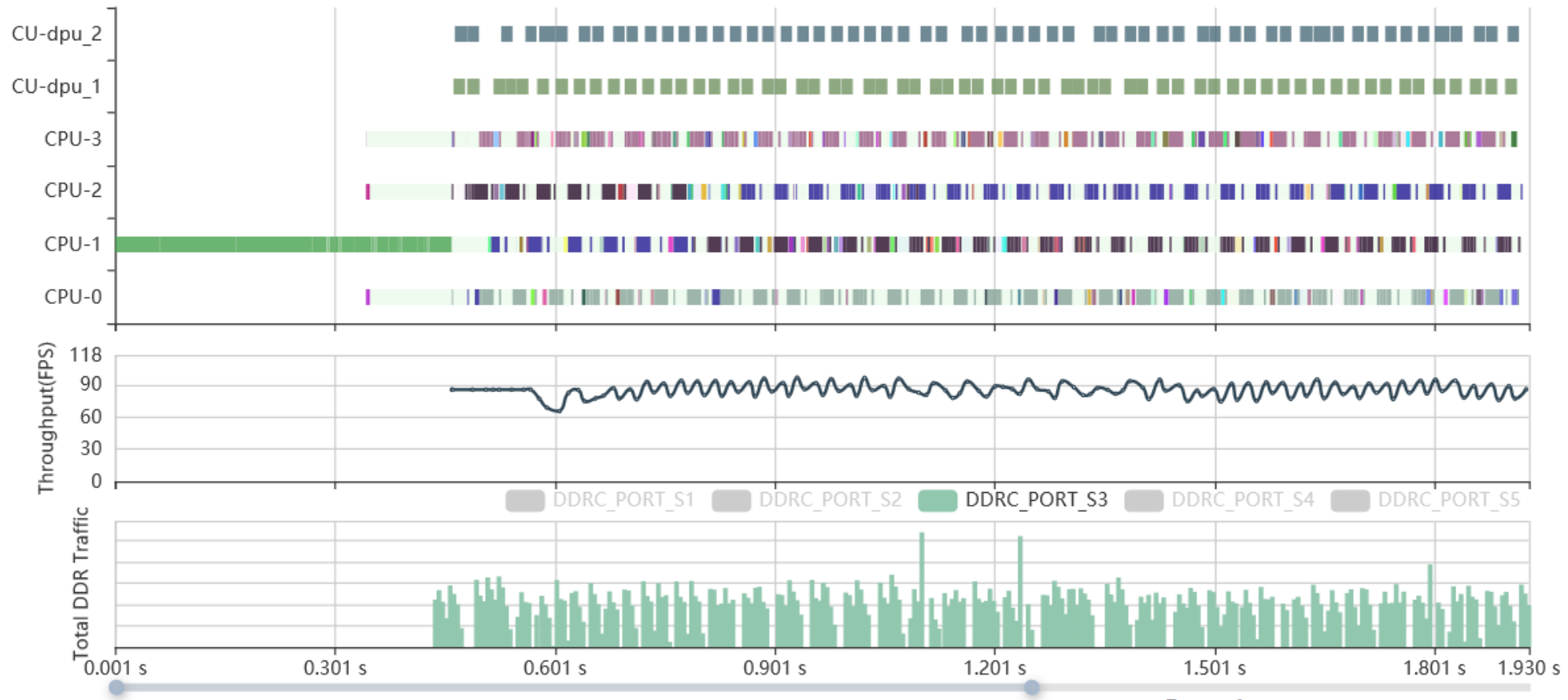
# AI Profiler

- ▶ The Vitis AI profiler tools is a set of tools that helps profile and visualize AI applications based on the Vitis AI Library:
  - Easy to use; requires neither the change in user's code nor re-compilation of the program
  - Figuring out hot spots or bottlenecks of preference at a glance
  - Illustrating the running state of difference computing units



# AI Profiler

## Xilinx Vitis AI Profile: Timeline



# DPU Target Reference Design

```
xdpu
├── dpu_ip          # rtl kernel
├── apps
│   └── Vitis
│       ├── models
│       ├── sample
│       ├── dnndk          # dnndk librarys
│       └── setup.sh
└── prj
    └── Vitis
        ├── kernel_xml    # pre-build SD card image
        │   ├── dpu
        │   └── sfm
        ├── Makefile
        ├── dpu_conf.vh
        ├── config_file   # config file
        │   ├── prj_config
        │   ├── prj_config_102_3dpu # integrate 3DPU on zcu102
        │   └── prj_config_104_2dpu # integrate 2DPU on zcu104
        ├── scripts
        └── README.md
```

> **DPU IP**

> **Model and related libraries**

> **Prebuild image**

> **Config files**

# Vitis AI



<https://github.com/Xilinx/Vitis-AI>

[https://github.com/Xilinx/Vitis-In-Depth-Tutorial/tree/master/Machine\\_Learning](https://github.com/Xilinx/Vitis-In-Depth-Tutorial/tree/master/Machine_Learning)



---

**Thank You**



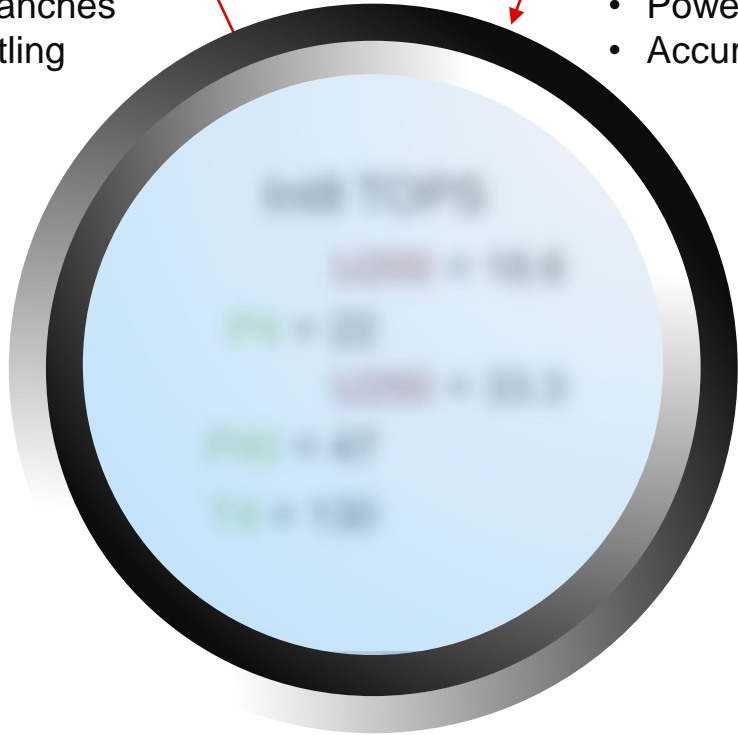
# Putting Metrics & Benchmarks in Focus

Usable TOPS limited by:

- Memory bottlenecks
- Code / data structure
- Stalls & branches
- Freq. throttling

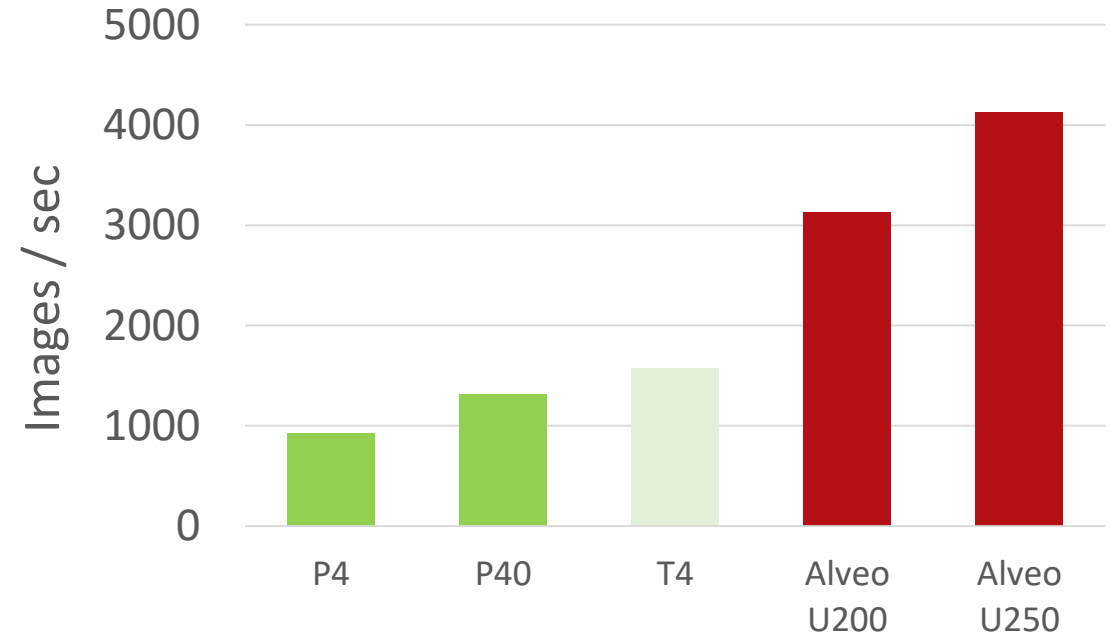
AND Application Reqs:

- Request ("Batch") Size
- Latency Spec
- Power Envelope
- Accuracy Requirements



## ML Benchmark

GoogleNetv1 Batch=1 Throughput



✓ Adaptable Hardware Delivers

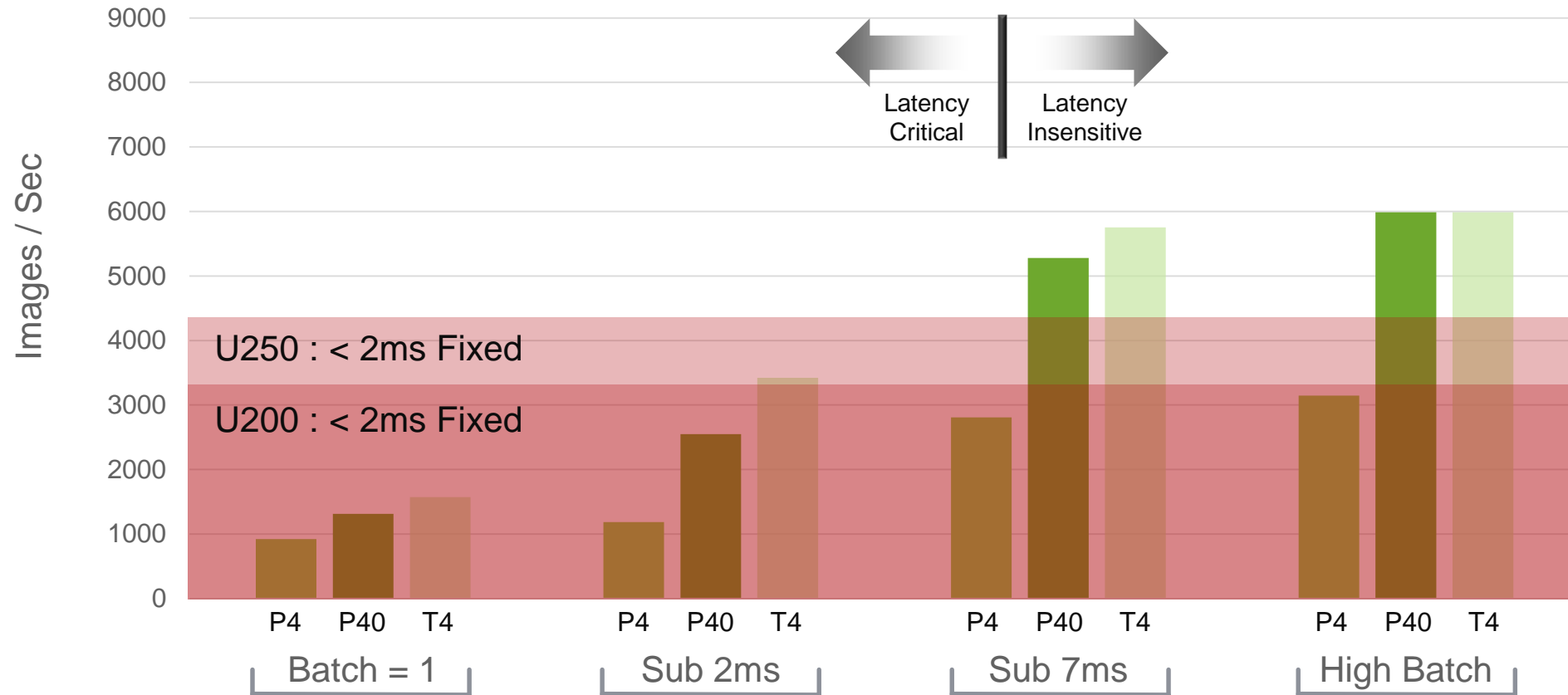
- Custom compute / memory
- Higher device utilization
- HW performance / power

Focus on Application Level Performance Where Xilinx Solutions Shine

# A Broader View of ML Benchmarking (Int8)

## Throughput vs. GPU Batch

GoogleNetv1 (Int8)

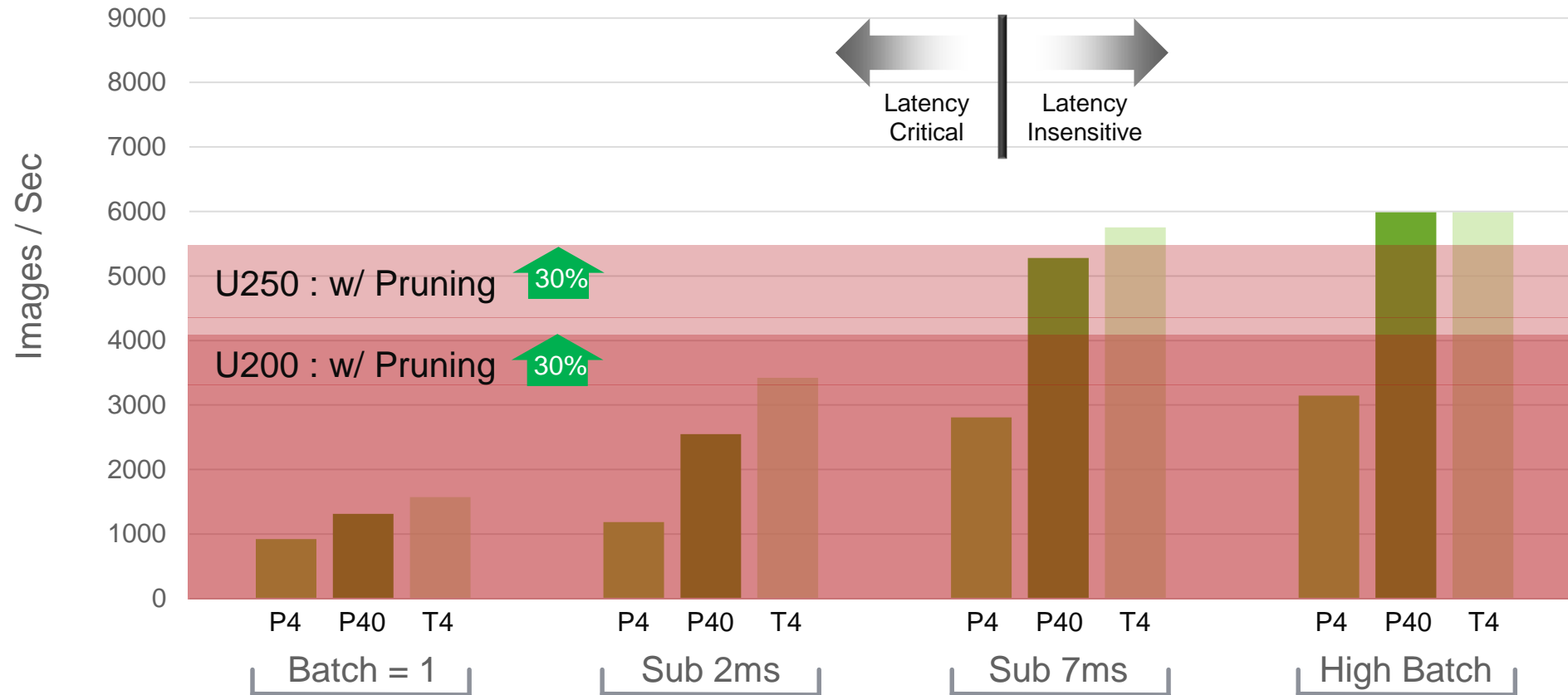


Alveo Delivers Low, Fixed Latency (< 2ms) in ALL Scenarios

# Next Step: Xilinx Pruning Techniques

## Throughput vs. GPU Batch

GoogleNetv1 (Int8)



DeepPhi Proprietary Pruning Increases Performance **30% or More**



# Xilinx vs. GPU: Edge ML Performance

(SSD 480x360)

