OpenVINO™ DEVCON Workshop Series 2023

Beyond the Continuum. The Importance of Quantization in Deep Learning

連続量の先へ

深層学習における量子化の重要性

Computer Vision Specialist

Technical Sales Specialist, Sales & Mktg Grp.

Performance optimization

Performance optimization

16.6 MB

4.7 MB

Quantization

„The process of constructing a discrete representation of a quantity that is usually regarded as continuous”

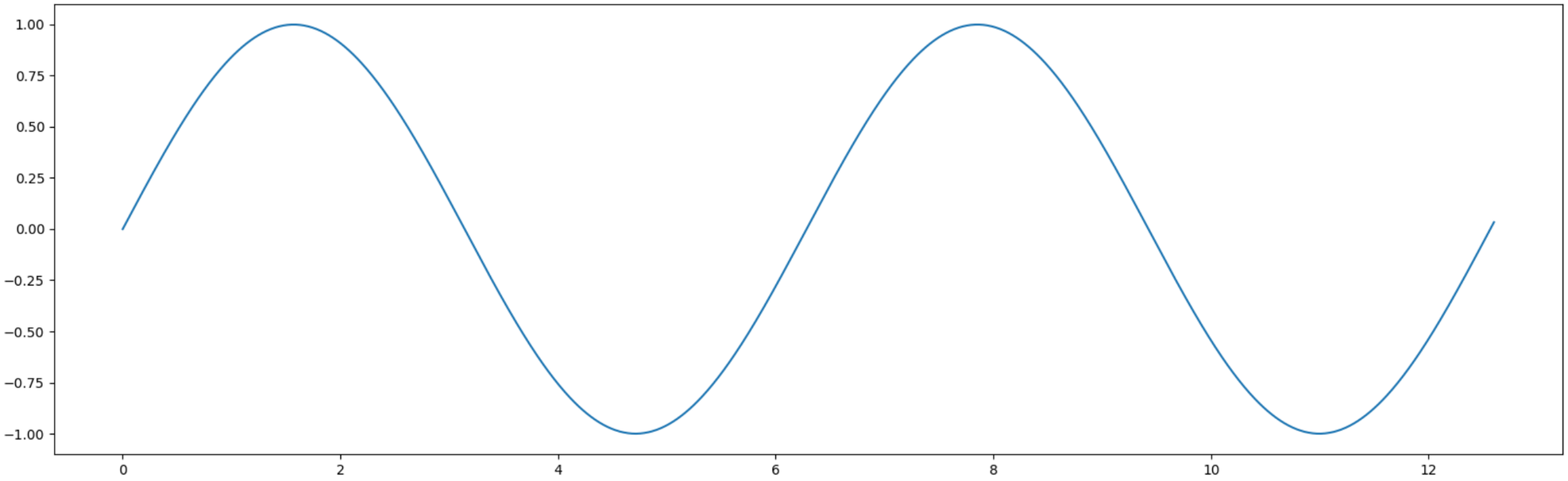
Encyclopedia.com

量子化 (物理学) - ある物理現象が、量子条件に合うような離散的な物理量をもつこと。古典力学の理論から量子力学の理論に移行するための手続きそのものを指す場合もある。

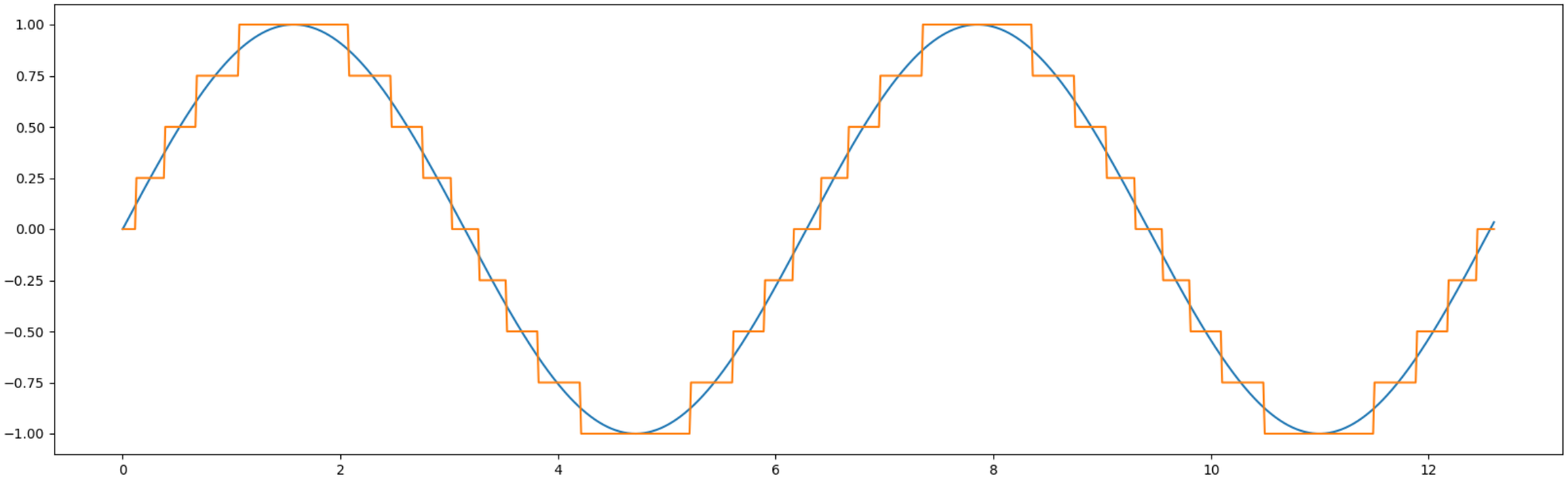
量子化 (情報科学) - 信号処理や画像処理において、信号の大きさを離散的な値で近似的に表すこと。

ja.wikipedia.org

Quantization

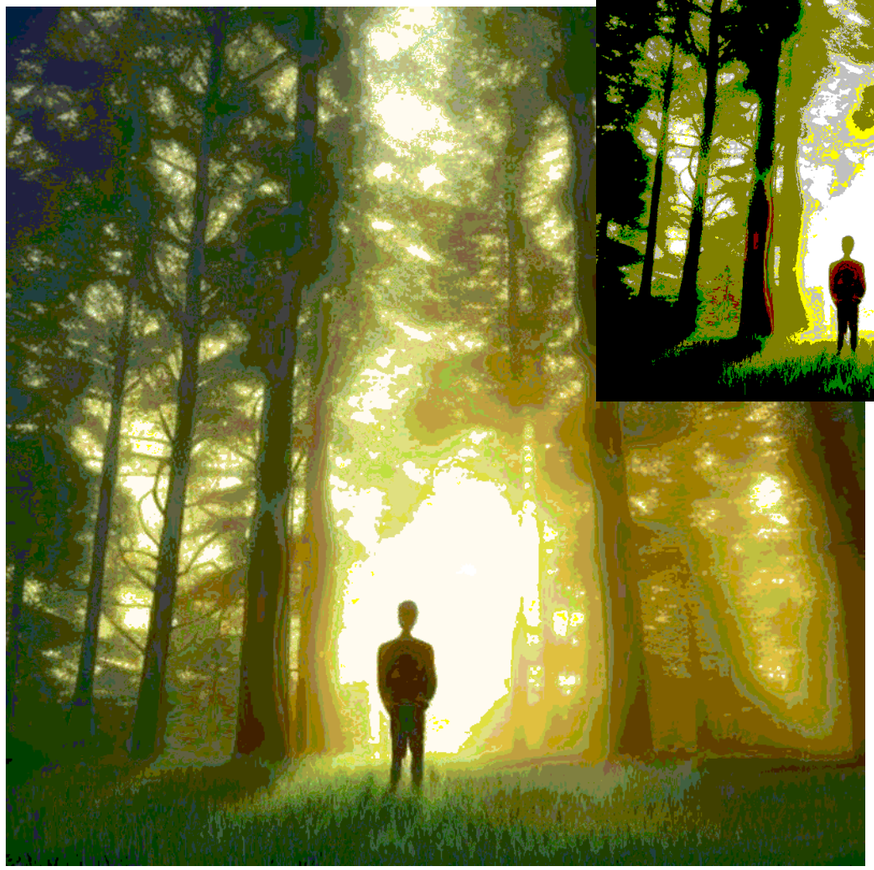


Quantization

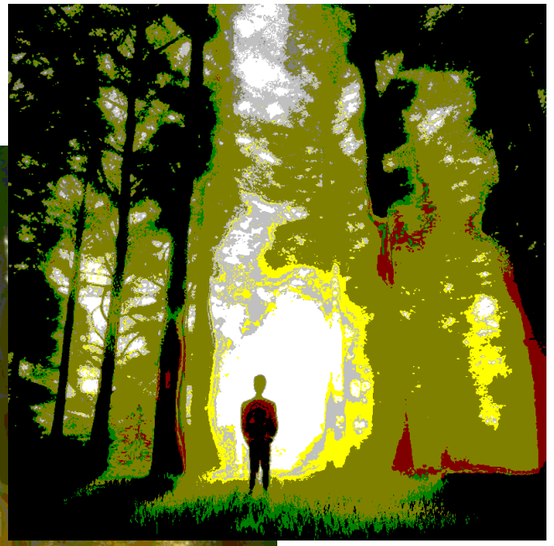


Color Quantization

16,777,216 colors = “FP32”



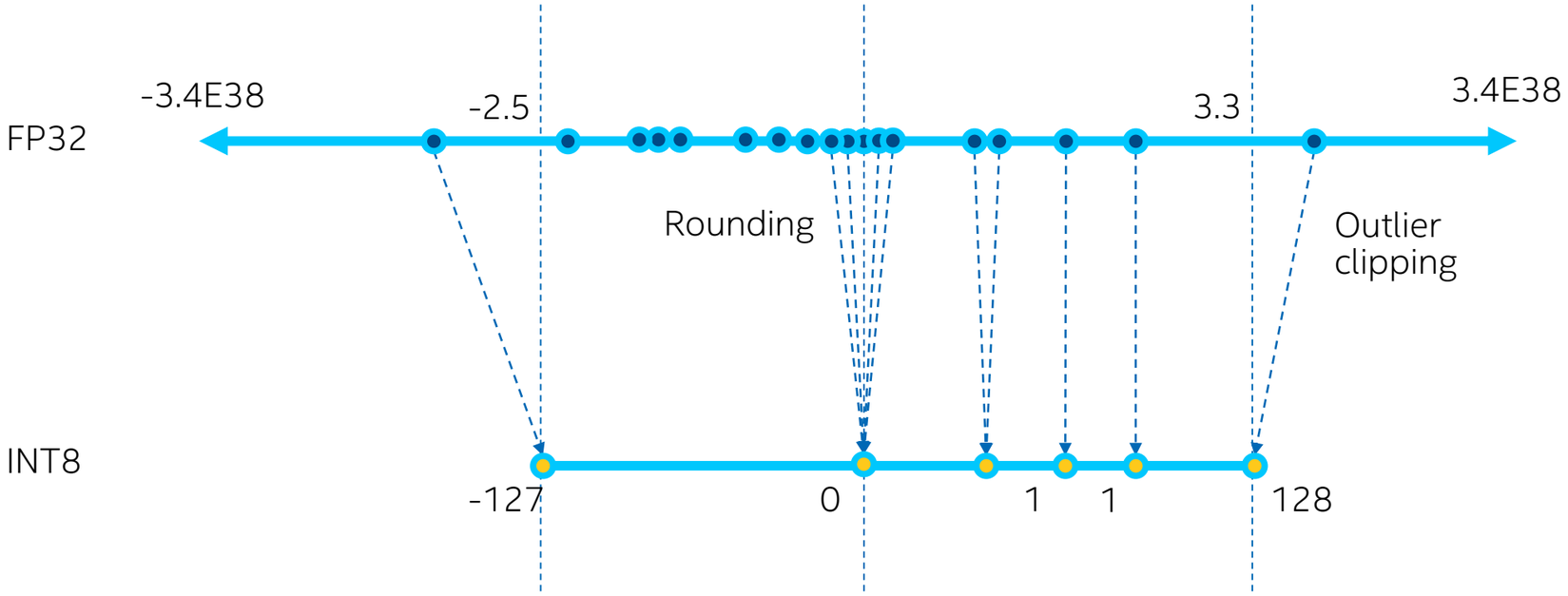
256 colors = “INT8”



16 colors
= “INT4”

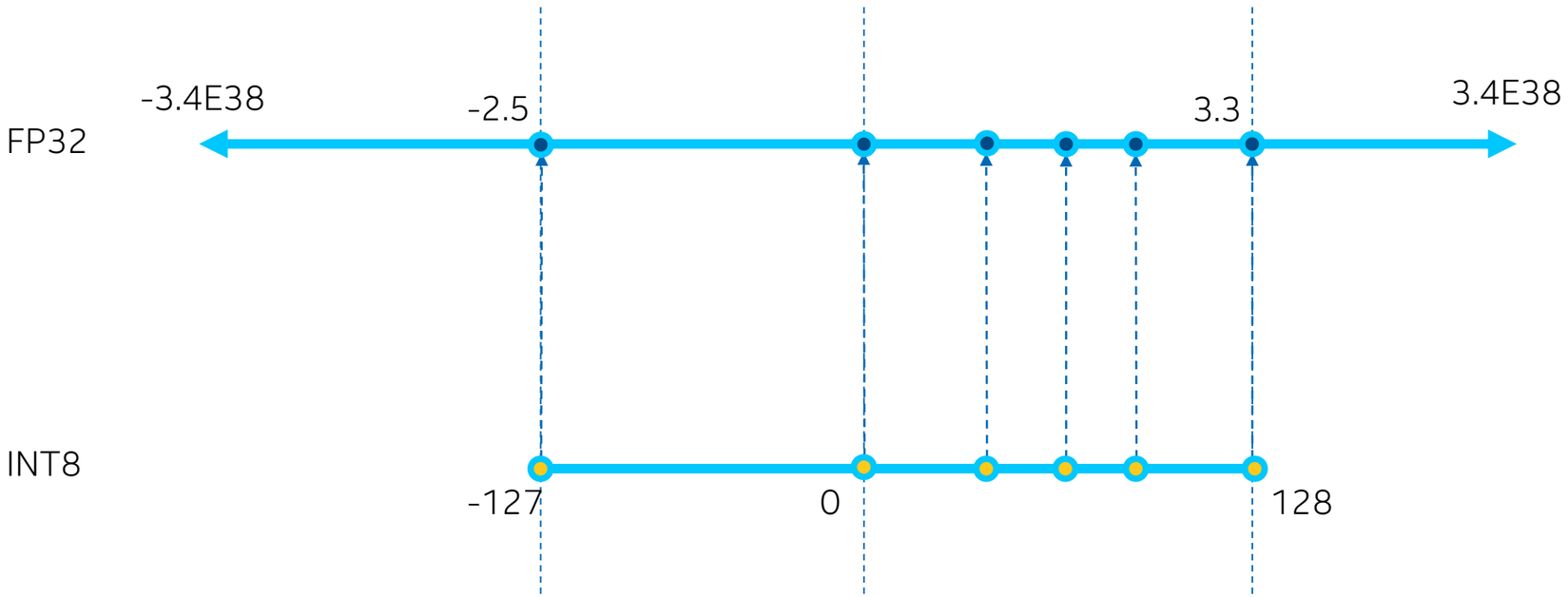
Quantization

宇宙の大きさ 138億光年 = 1.12×10^{26}m




```
int8_value = round(real_value / scale) + zero_point
```

Dequantization



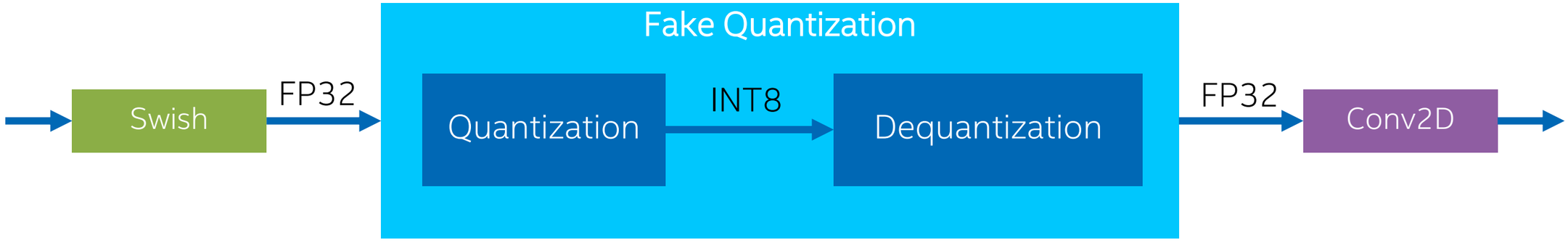

```
real_value = (int8_value - zero_point) * scale
```

Quantization Types

Fake Quantization

量子化したときと同様の離散値を
取る浮動小数点データ

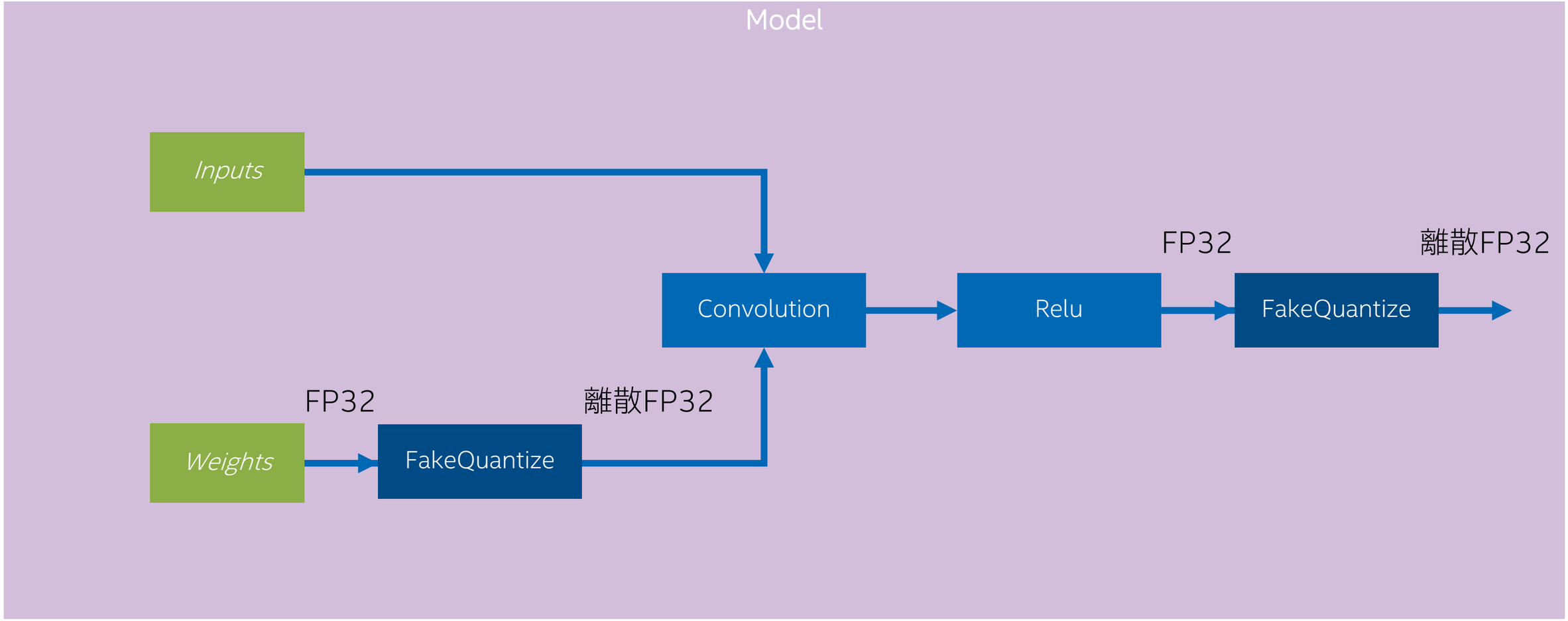
1.2, 4.7, 2.3　　1, 4, 2　　1.0, 4.0, 2.0



* 実際にはscaleとoffsetが絡んでくるので、この例のように単なる小数点以下切り捨てとは異なる結果になります。

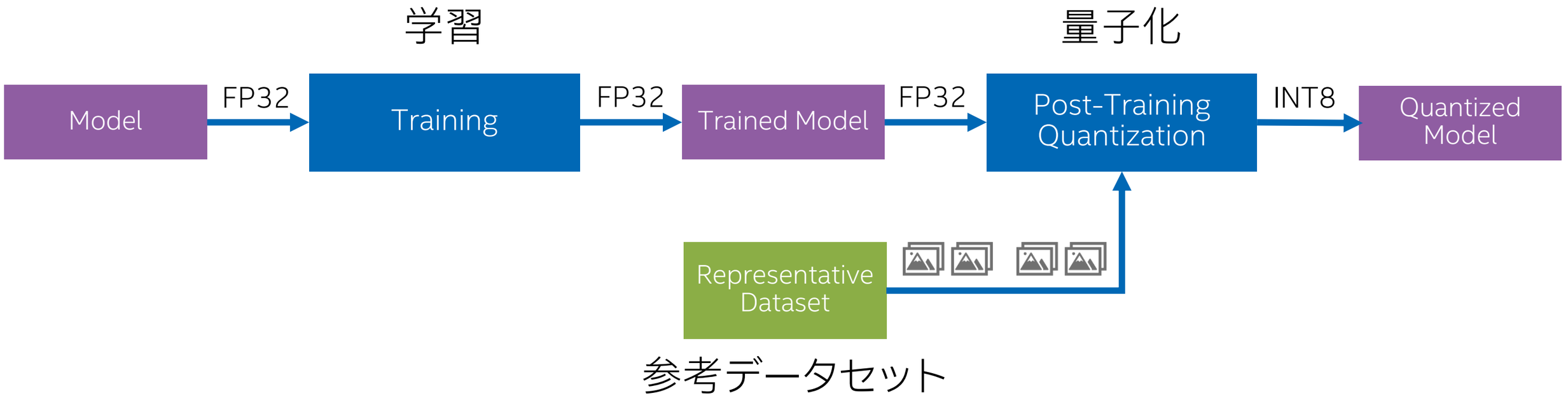
Quantization-Aware Training (QAT)

量子化を考慮した学習

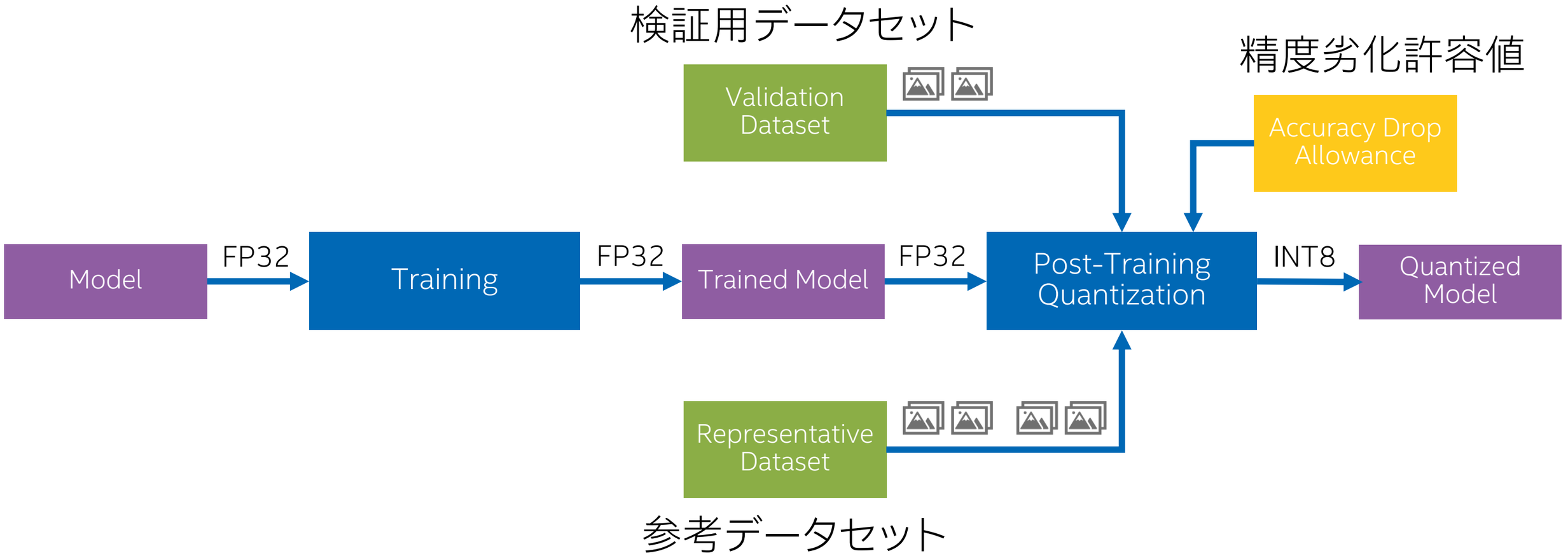


Post-Training Quantization (PTQ)

学習後量子化



Accuracy-Control Quantization



OpenVINO + NNCF

OpenVINO™ OPTIMIZATION TOOLS

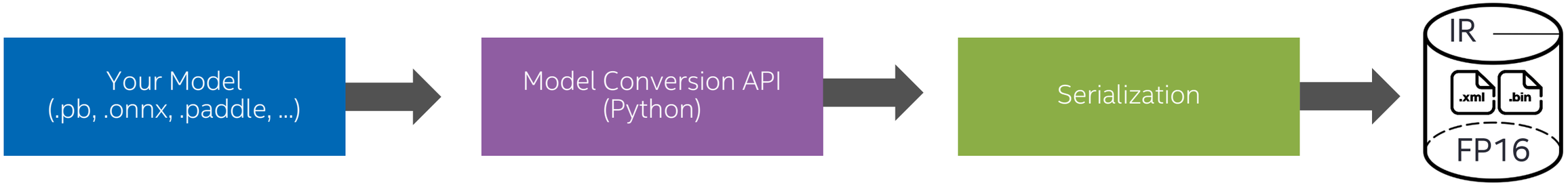
MO

Model Optimizer API

NNCF

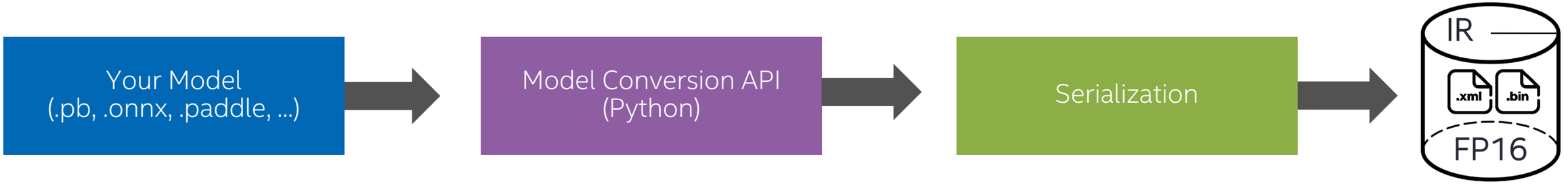
Neural Network Compression Framework

Model Conversion API



For workloads and configurations visit www.intel.com/PerformanceIndex. Results may vary.

Model Conversion API



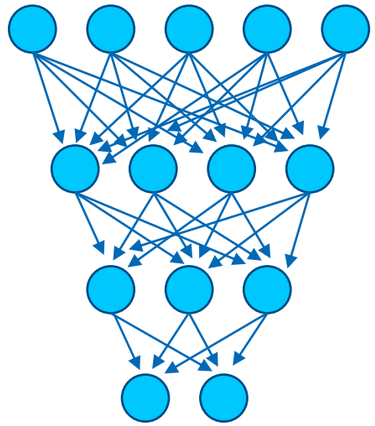

```
ov_model = ov.convert_model("model.onnx",
                            input_shape=[1,3,-1,-1],
                            compress_to_fp16=True)

ov.serialize(ov_model, "converted_model.xml")
```

For workloads and configurations visit www.intel.com/PerformanceIndex. Results may vary.

Neural Network Compression Framework (NNCF)

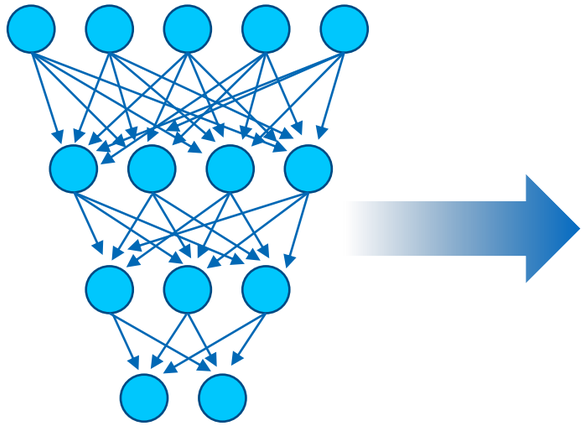
TensorFlow PyTorch

OpenVINO™ ONNX



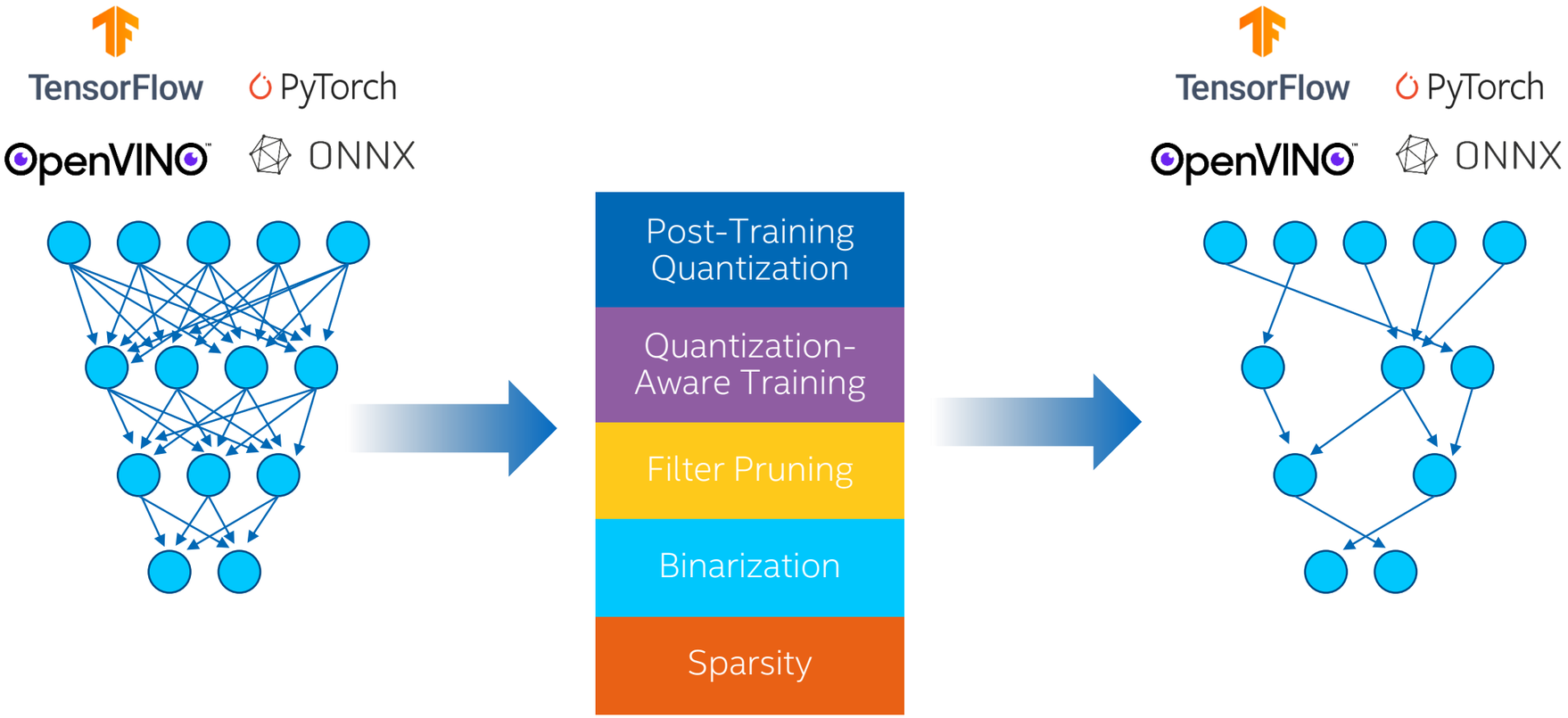
Neural Network Compression Framework (NNCF)

TensorFlow PyTorch
OpenVINO ONNX

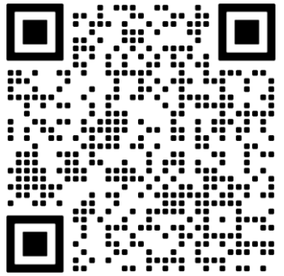


Post-Training Quantization	学習後量子化
Quantization-Aware Training	量子化を考慮した学習
Filter Pruning	フィルタ・プルーニング
Binarization	2値化
Sparsity	スパース化

Neural Network Compression Framework (NNCF)



OpenVINO™ Runtime



```python
from openvino import runtime as ov

img = load_img()

core = ov.Core()

model = core.read_model(model="model.xml")
compiled_model = core.compile_model(model=model, device_name="CPU")

output_layer = compiled_model.outputs[0]

result = compiled_model(img)[output_layer]
```

Post-Training Quantization (NNCF)

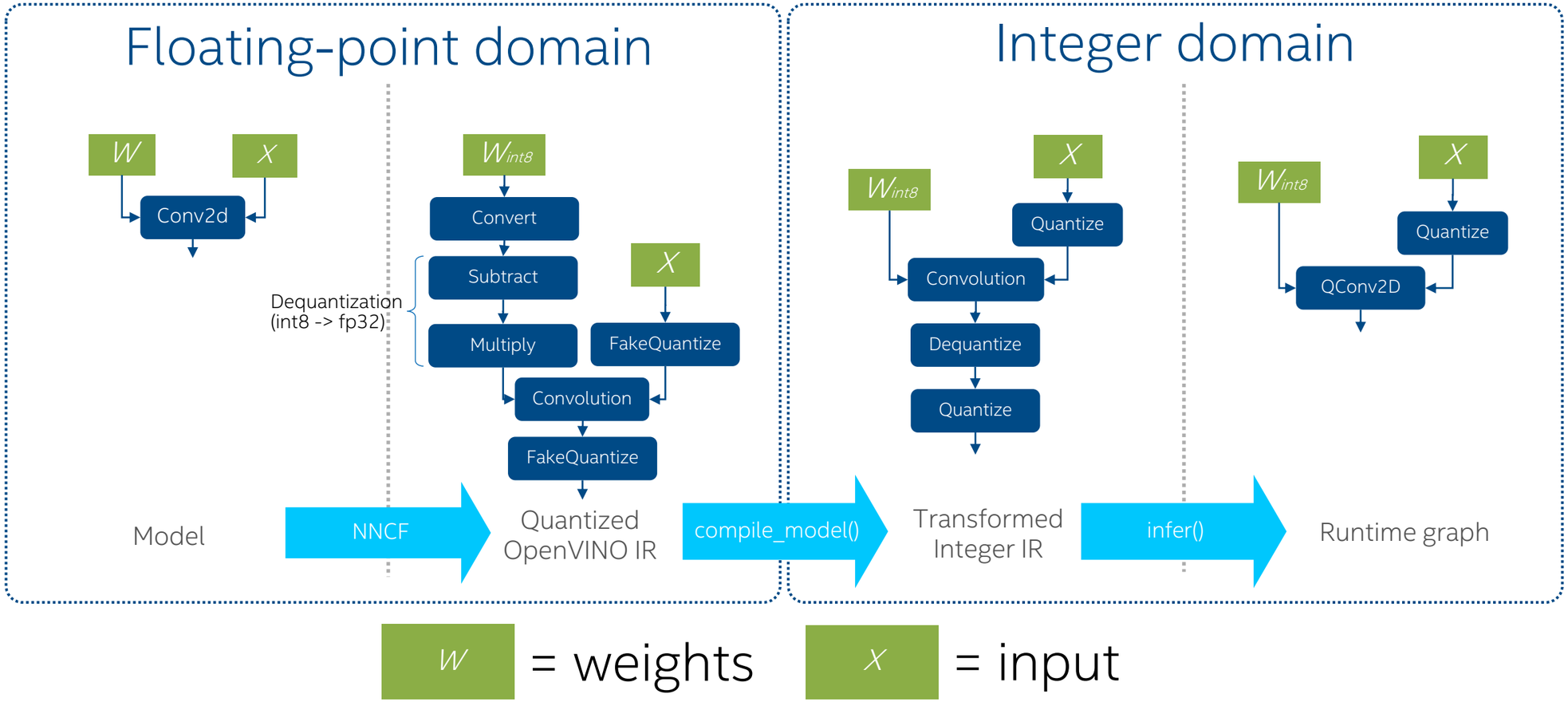
1. Prepare a Calibration Dataset

```
import nncf

calibration_loader = torch.utils.data.DataLoader(...)

def transform_fn(data_item):
    images, _ = data_item
    return images

calibration_dataset = nncf.Dataset(calibration_loader, transform_fn)
```

2 . Run a Quantized Model

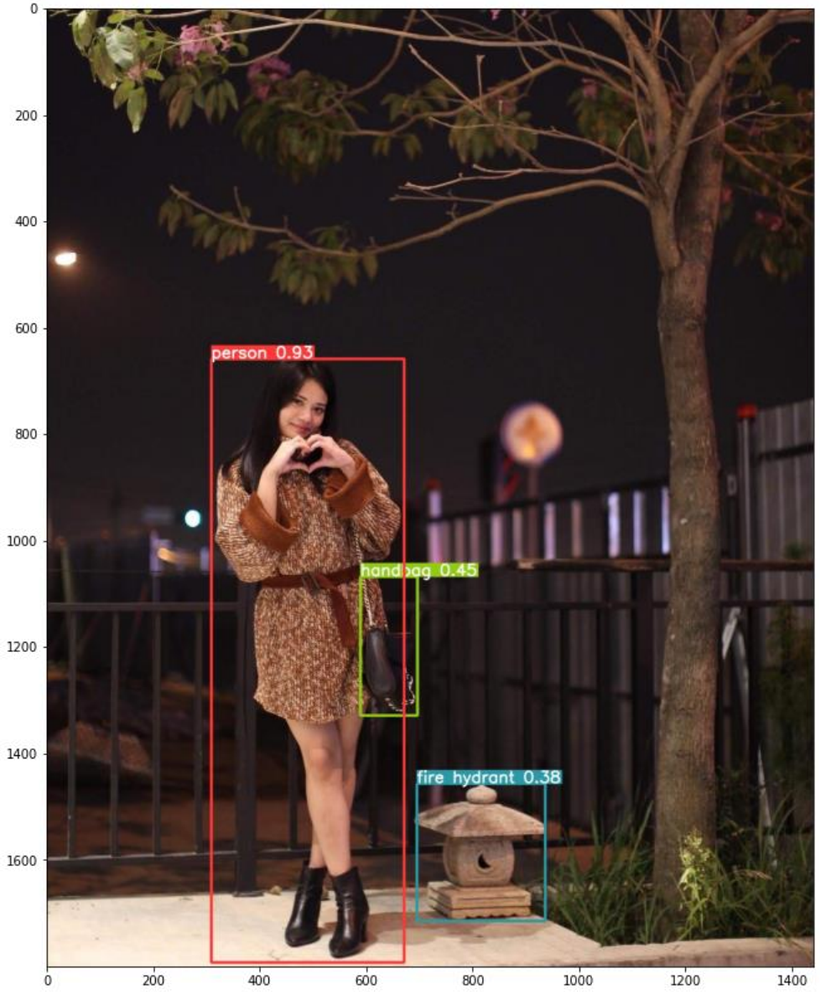
```
model = ... # OpenVINO/ONNX/PyTorch/TF object

quantized_model = nncf.quantize(model, calibration_dataset)
```

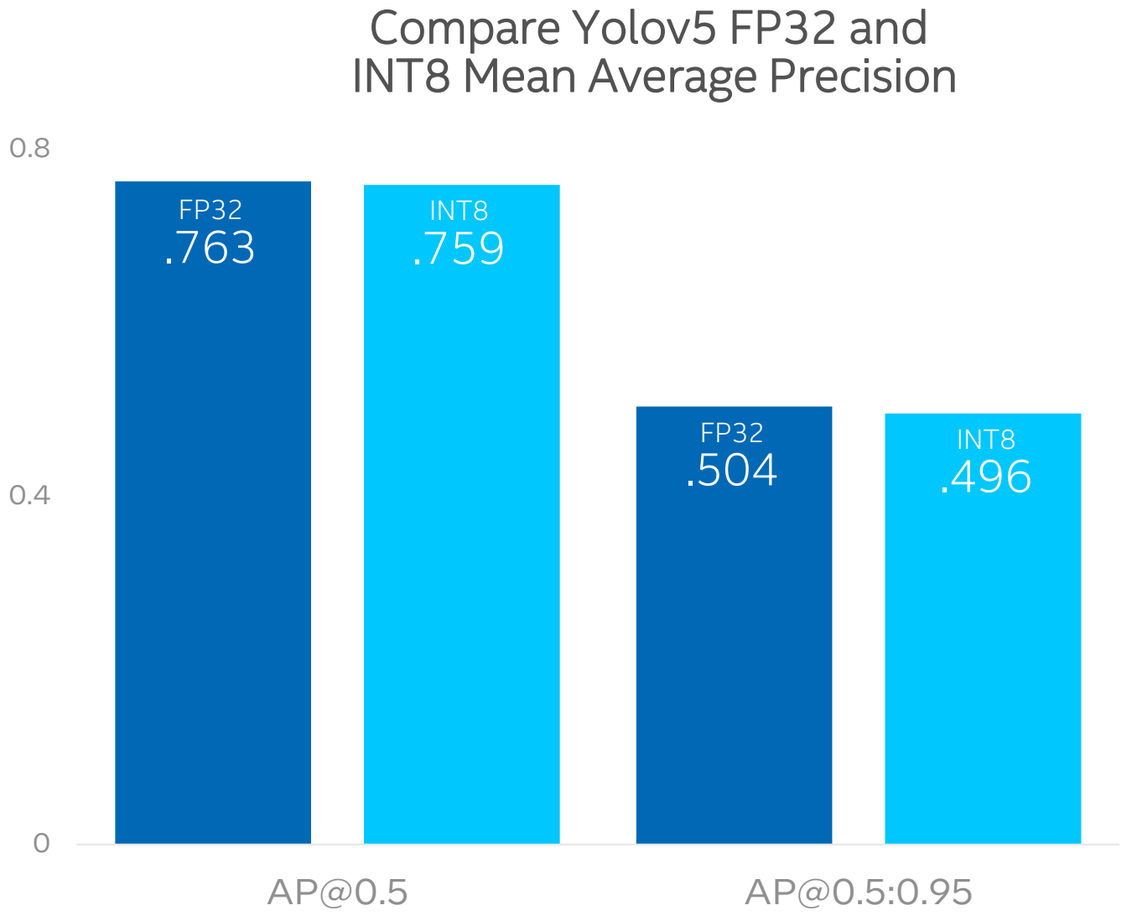
Stages of the Model (OpenVINO + NNCF)



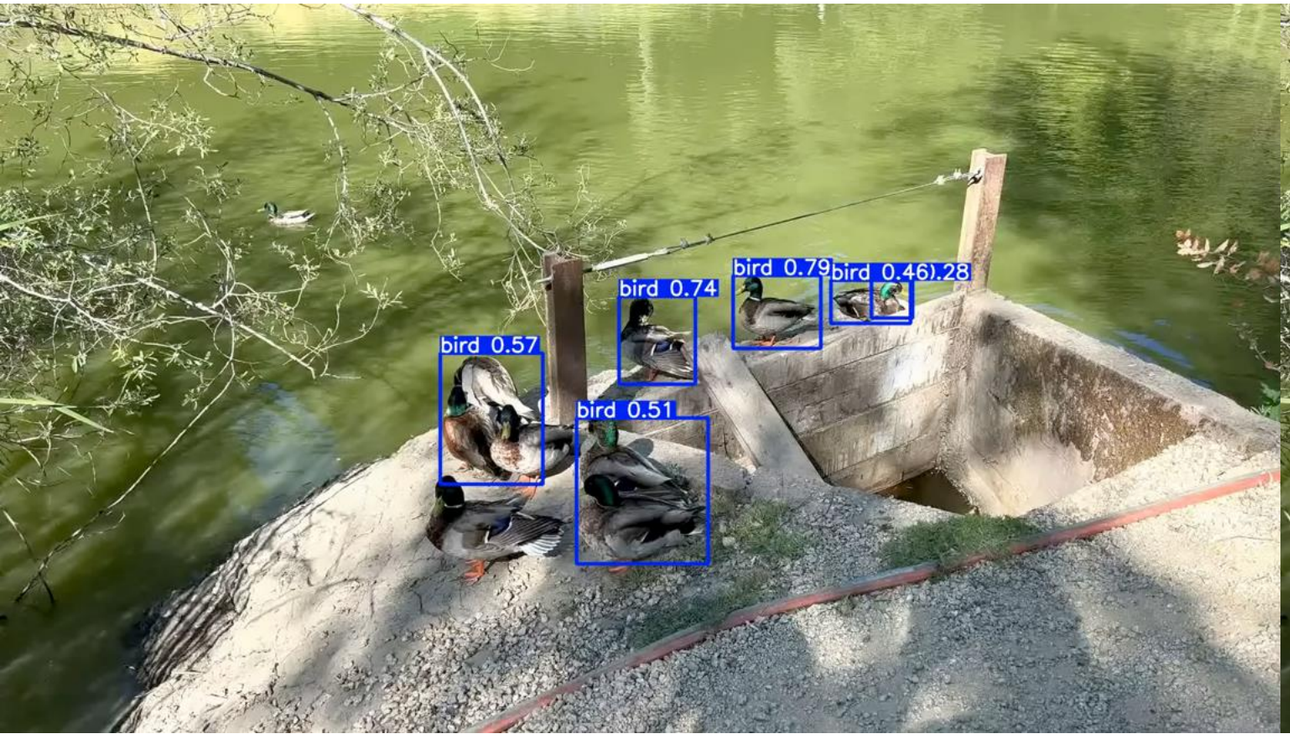
W = weights X = input

Post-Training Quantization (NNCF)

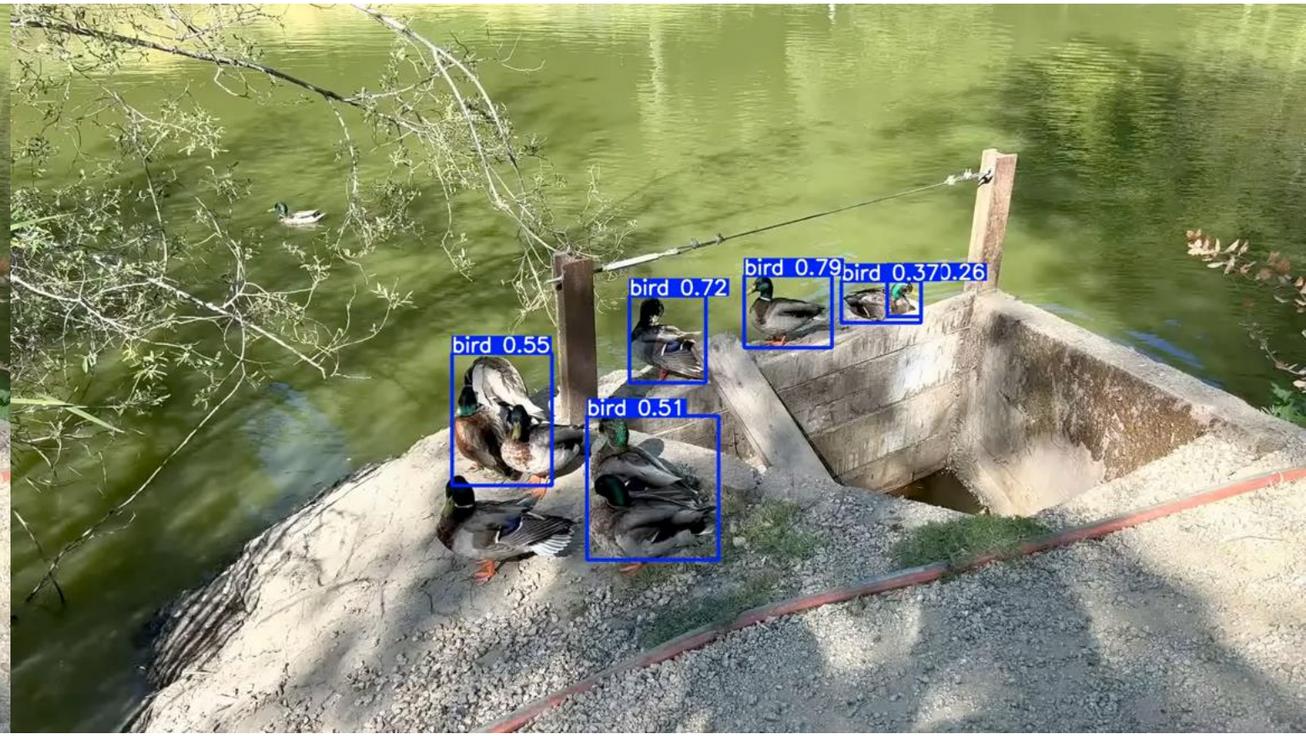




Quantization Results Comparison (YOLOv5)

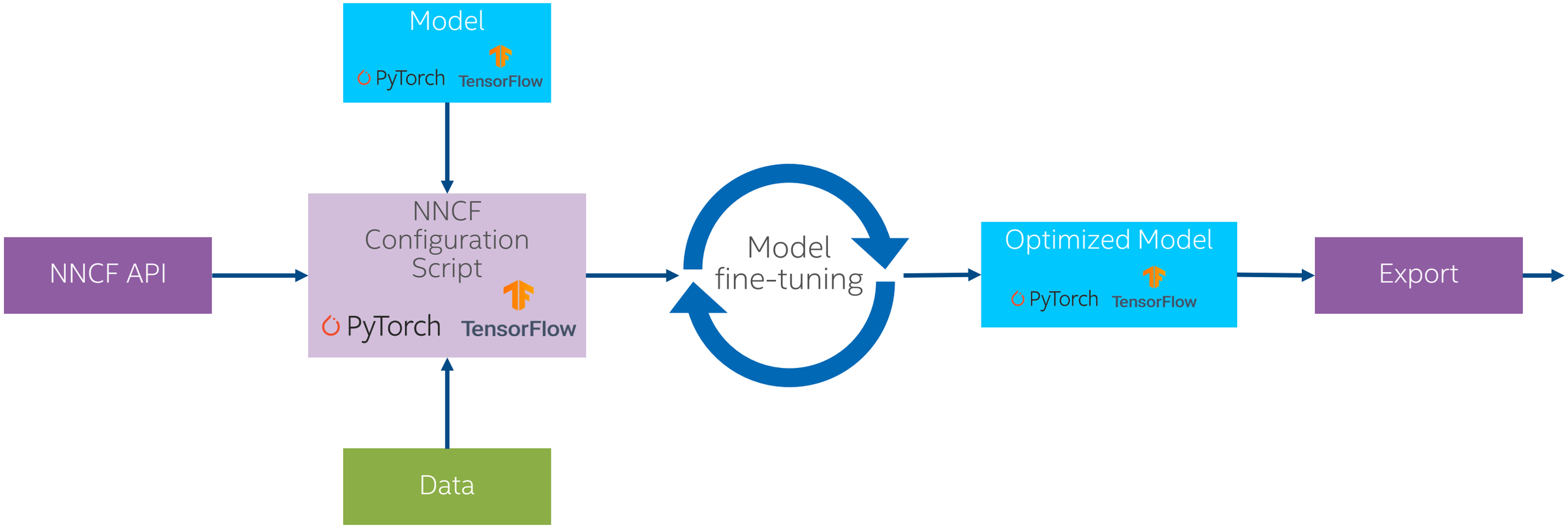


FP32

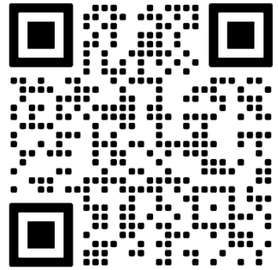


INT8

Quantization-Aware Training (NNCF)



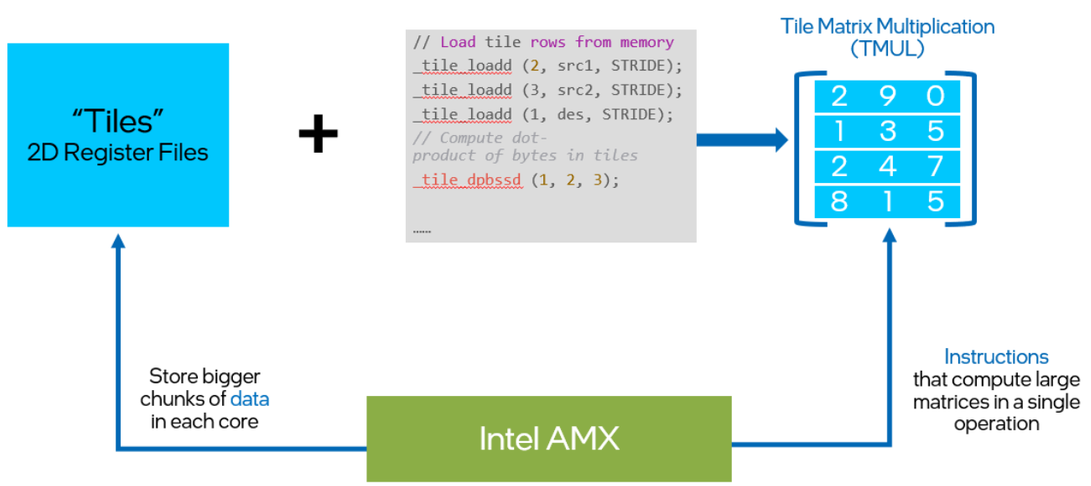
Quantized Stable Diffusion with Optimum-Intel

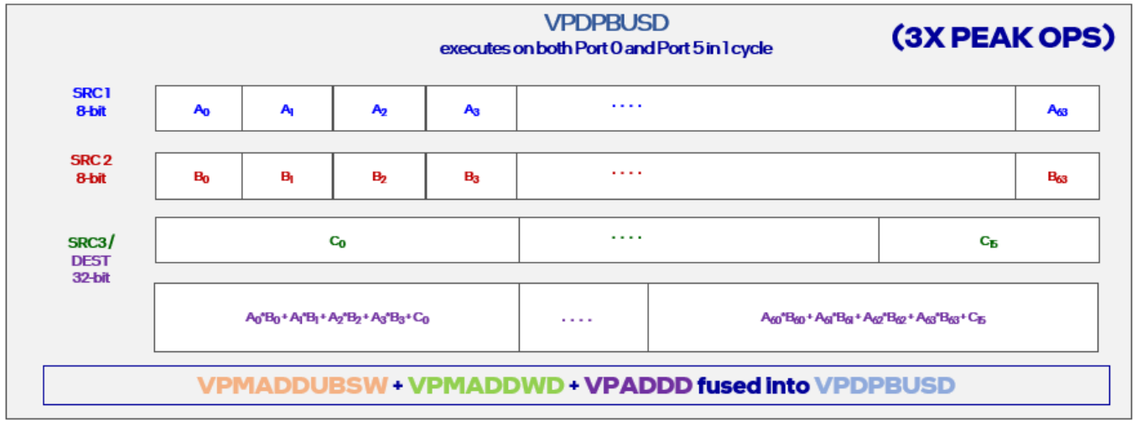


```
from optimum.intel.openvino import OVStableDiffusionPipeline

pipe = OVStableDiffusionPipeline\
       .from_pretrained("OpenVINO/stable-diffusion-2-1-quantized", compile=False)
pipe.reshape(batch_size=1, height=512, width=512, num_images_per_prompt=1)
pipe.compile()

prompt = "sailing ship in storm by Rembrandt"
output = pipe(prompt, num_inference_steps=50, output_type="pil")
output.images[0].save("result.png")
```

Quantized Stable Diffusion with Optimum-Intel

Boosting Performance of Quantized Model with 4th Gen Intel® Xeon®





VNNI

Let’s Run the Live Code!

Quantization with NNCF

Post-Training Quantization

https://github.com/openvinotoolkit/nncf/tree/develop/examples/post_training_quantization/openvino/yolov8

Accuracy-Control Quantization

https://github.com/openvinotoolkit/nncf/tree/develop/examples/post_training_quantization/openvino/yolov8_quantize_with_accuracy_control

Quantization-Aware Training

https://github.com/openvinotoolkit/openvino_notebooks/tree/main/notebooks/302-pytorch-quantization-aware-training

www.openvino.ai

https://github.com/openvinotoolkit/openvino_notebooks/wiki/Connect-with-us

Connect With Us

Notices and disclaimers

Intel is committed to respecting human rights and avoiding complicity in human rights abuses. See Intel's Global Human Rights Principles. Intel® products and software are intended only to be used in applications that do not cause or contribute to a violation of an internationally recognized human right.

Intel® technologies may require enabled hardware, software, or service activation. No product or component can be absolutely secure. Your costs and results may vary.