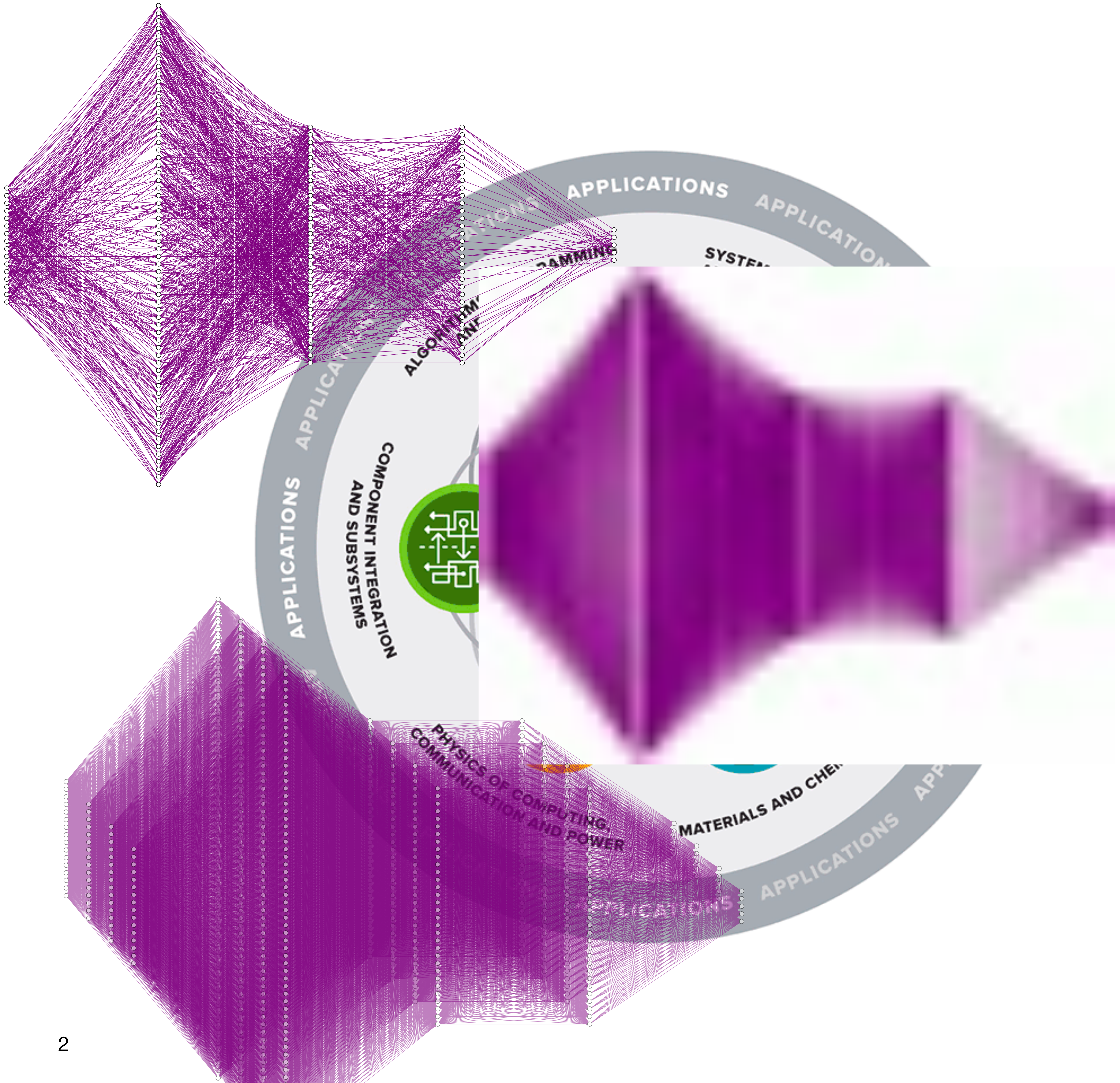
PHYS 139/239: Machine Learning in Physics

Lecture 15: Knowledge distillation

Javier Duarte — February 28, 2023

Recap: Codesign

- **Codesign**: intrinsic development loop between algorithm design, training, and implementation
- Compression
  - Maintain high performance while removing redundant operations
- Quantization
  - Reduce precision from 32-bit floating point to 16-bit, 8-bit, …
- Parallelization
  - Balance parallelization (how fast) with resources needed (how costly)



Recap: Quantization types

arXiv:2004.09602

- Quantization: using reduced precision for parameters and operations
- Fixed-point precision

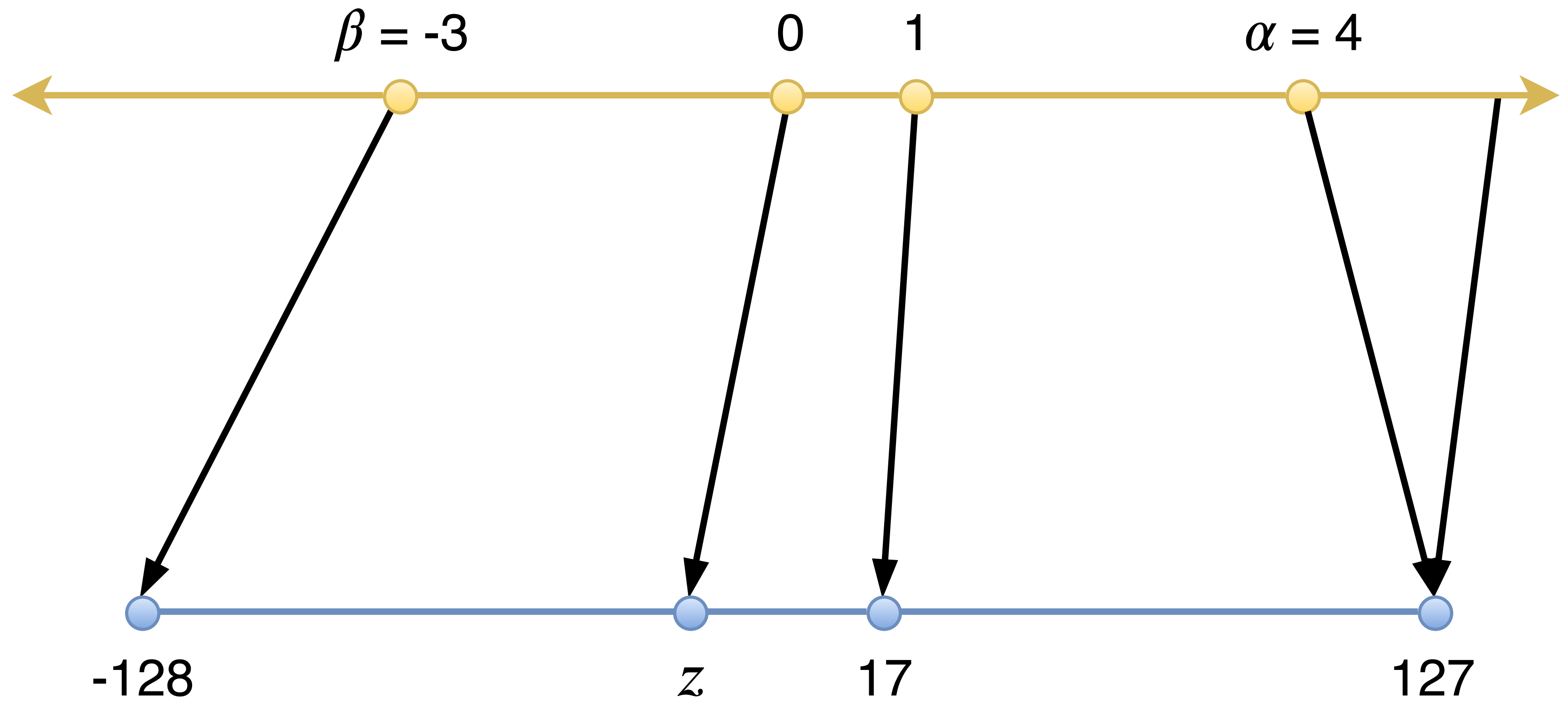
ap_fixed<width,integer>

`0101.1011101010`

integer fractional

width

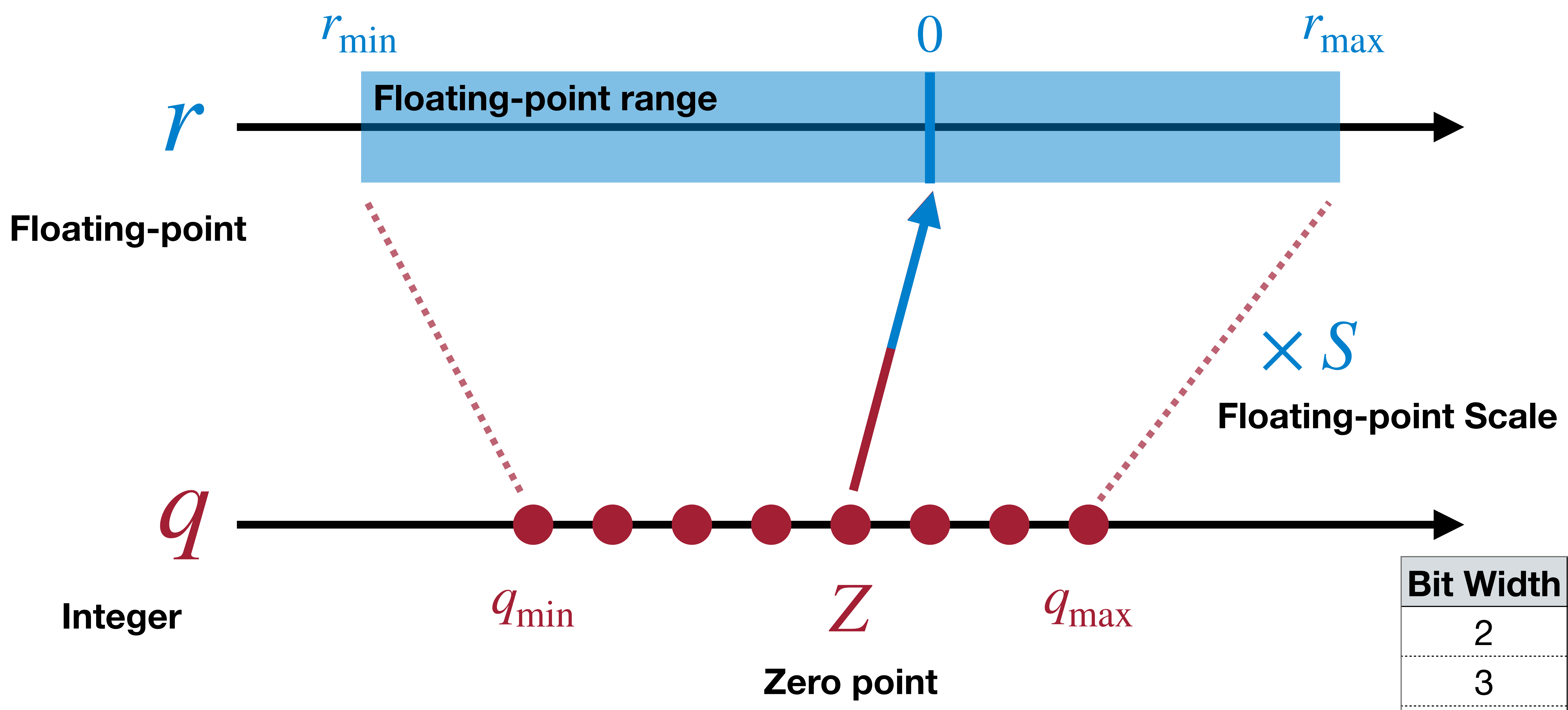
- Affine integer quantization



Affine integer quantization

arXiv:2004.09602

An affine mapping of integers to real numbers $r = S(q - Z)$



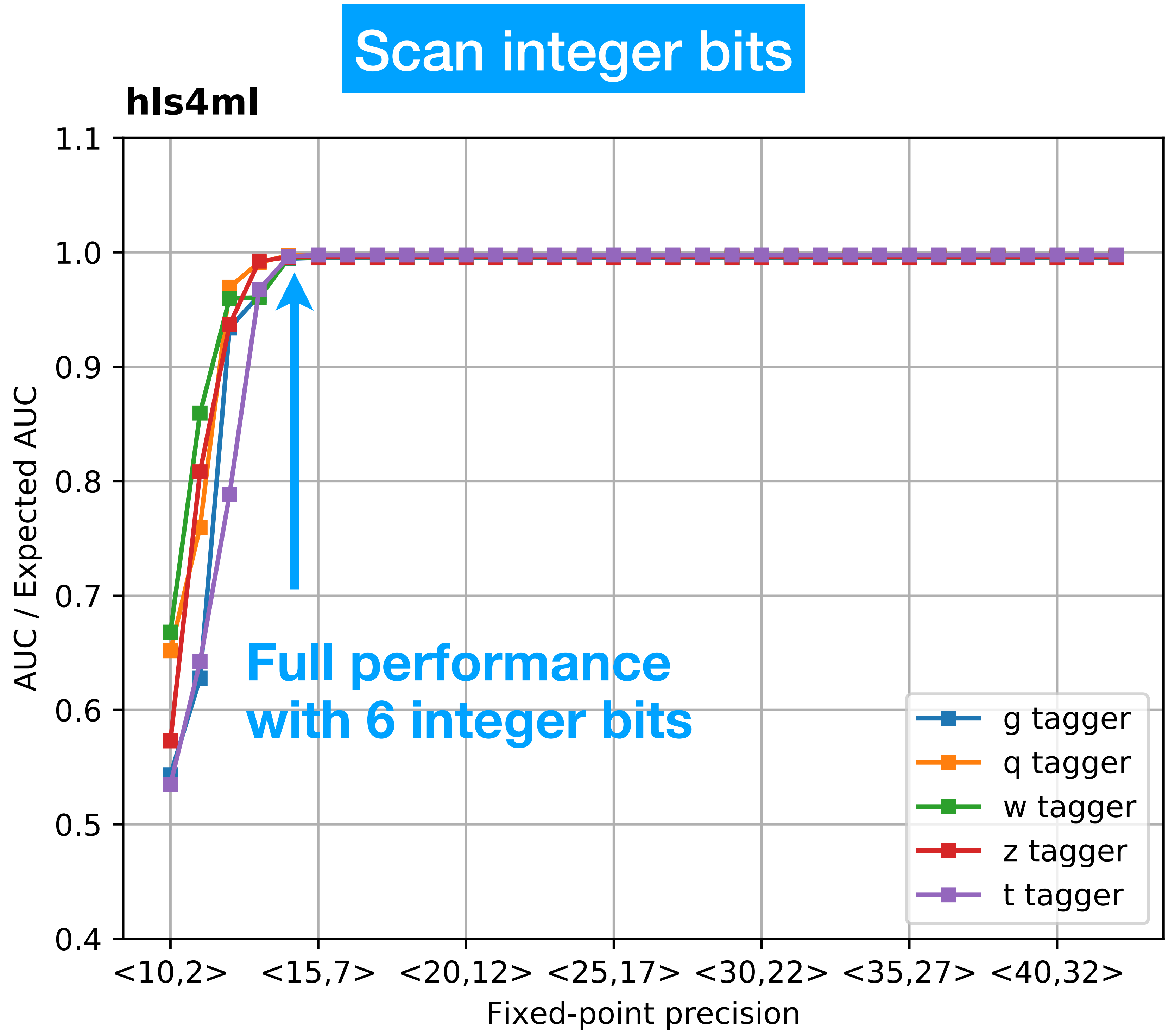
Bit Width	q_{min}	q_{max}
2	-2	1
3	-4	3
4	-8	7
N	-2^{N-1}	$2^{N-1}-1$

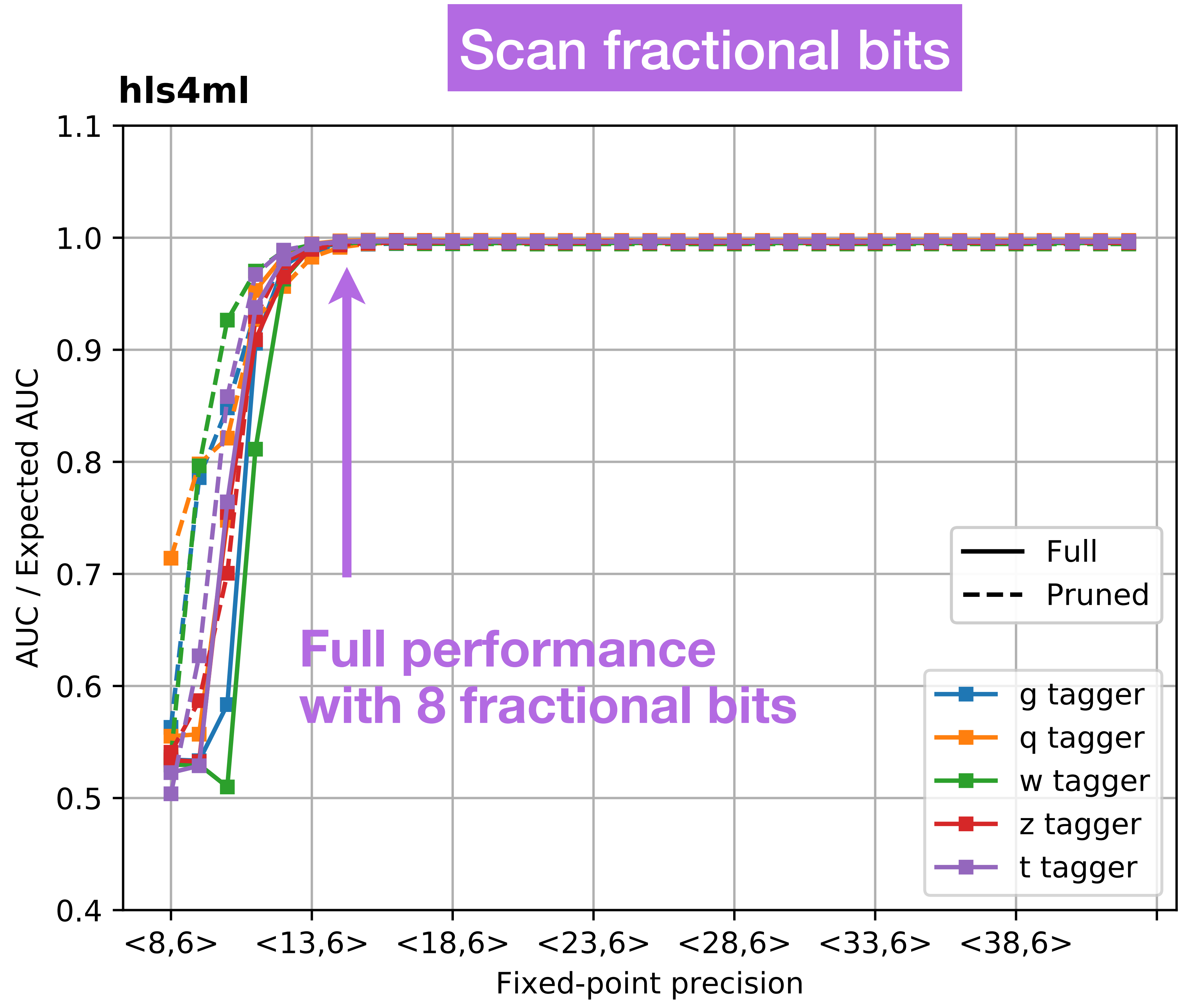
Quantization and Training of Neural Networks for Efficient Integer-Arithmetic-Only Inference [Jacob *et al.*, CVPR 2018]

Post-training quantization vs. quantization-aware training

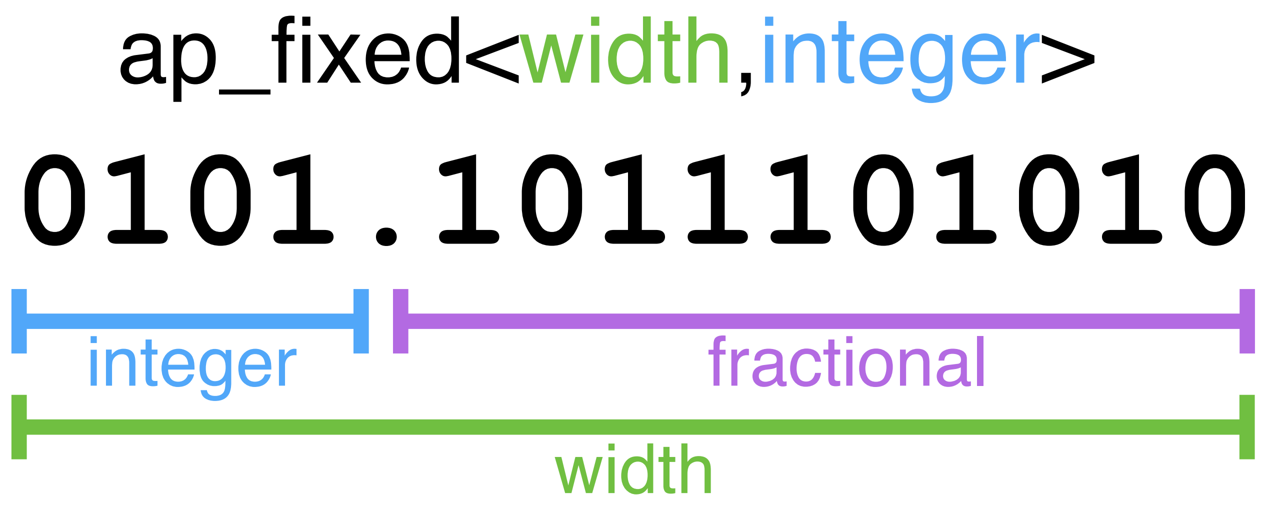
PTQ	QAT
Usually fast	Slow
No re-training of the model	Model needs to be trained/finetuned
Plug and play of quantization schemes	Plug and play of quantization schemes (requires re-training)
Less control over final accuracy of the model	More control over final accuracy since *q-params* are learned during training.

Post-training quantization





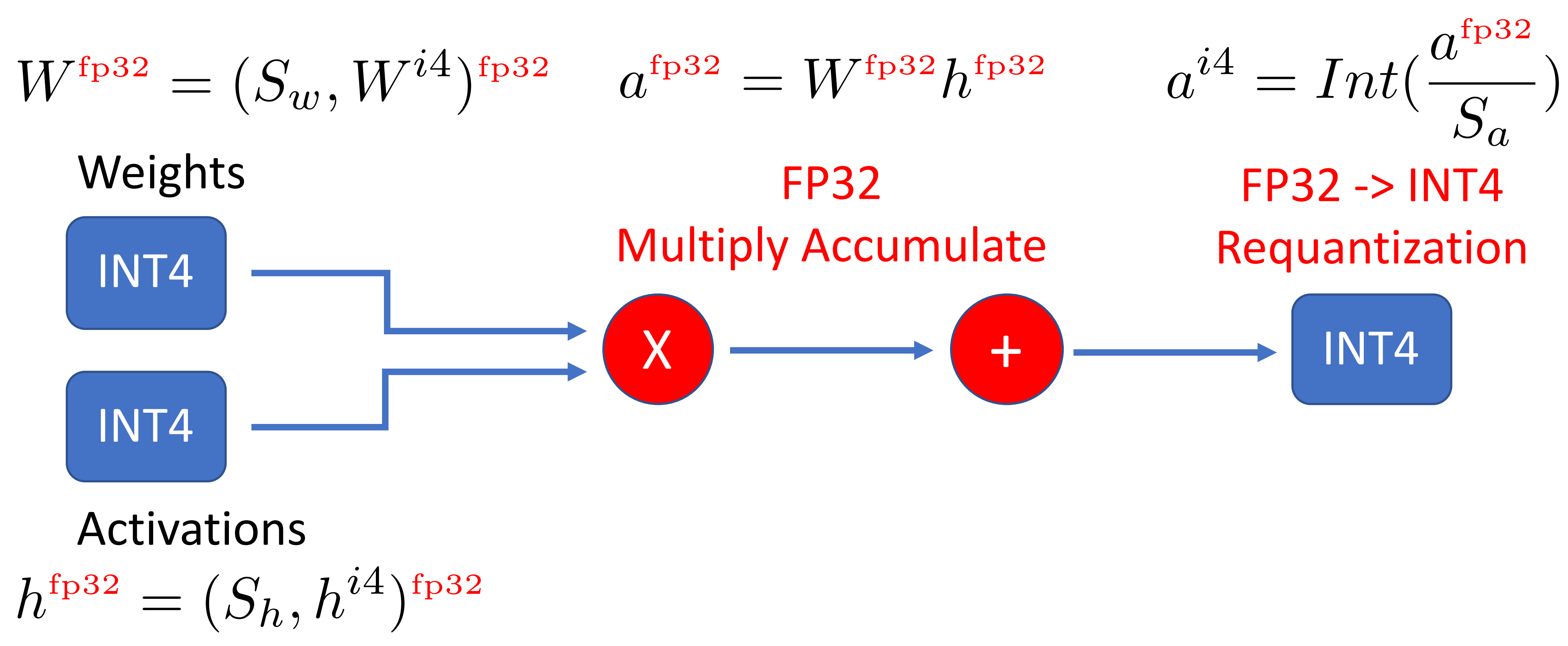
- General strategy: avoid overflows in integer bit then scan the decimal bit until reaching optimal performance

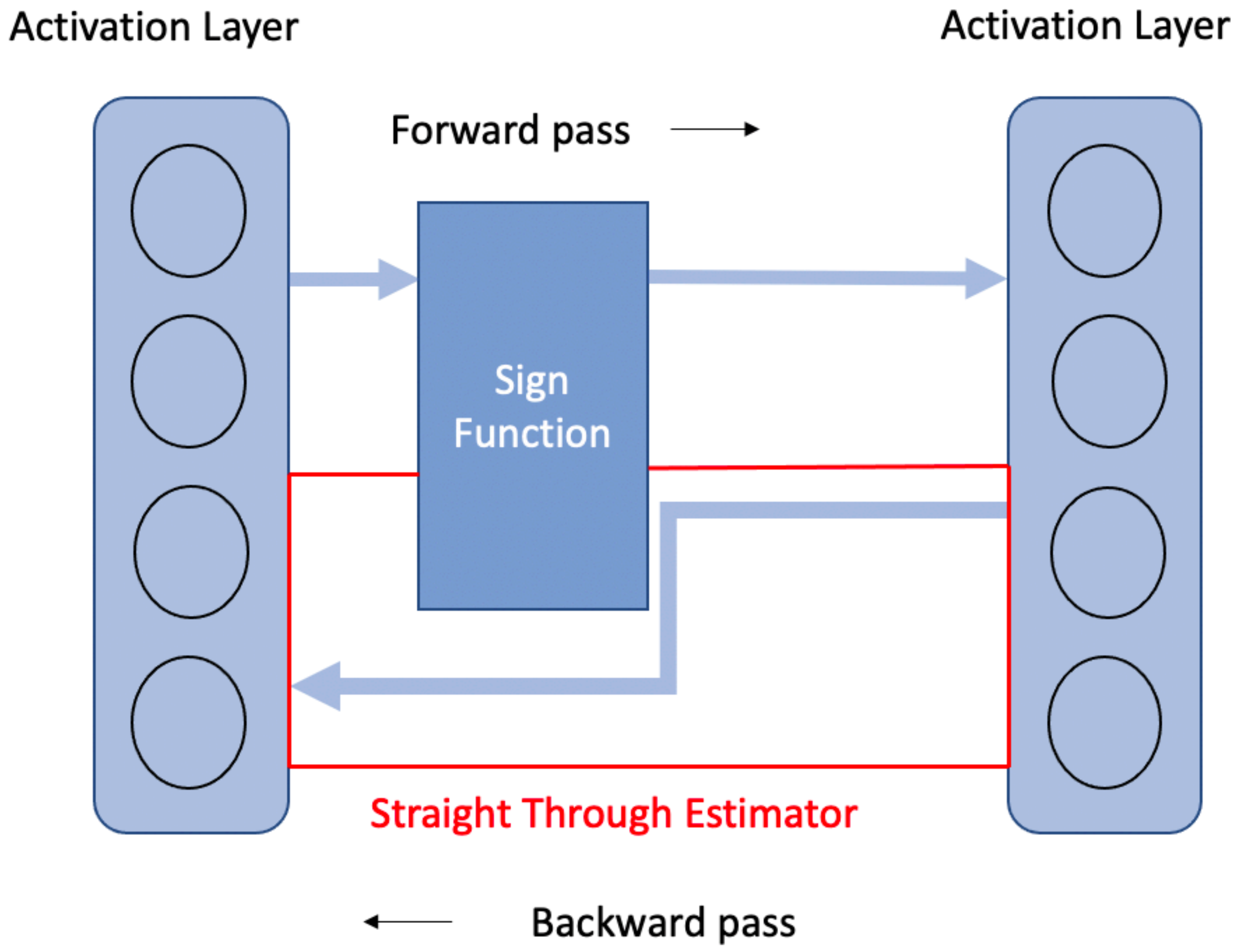


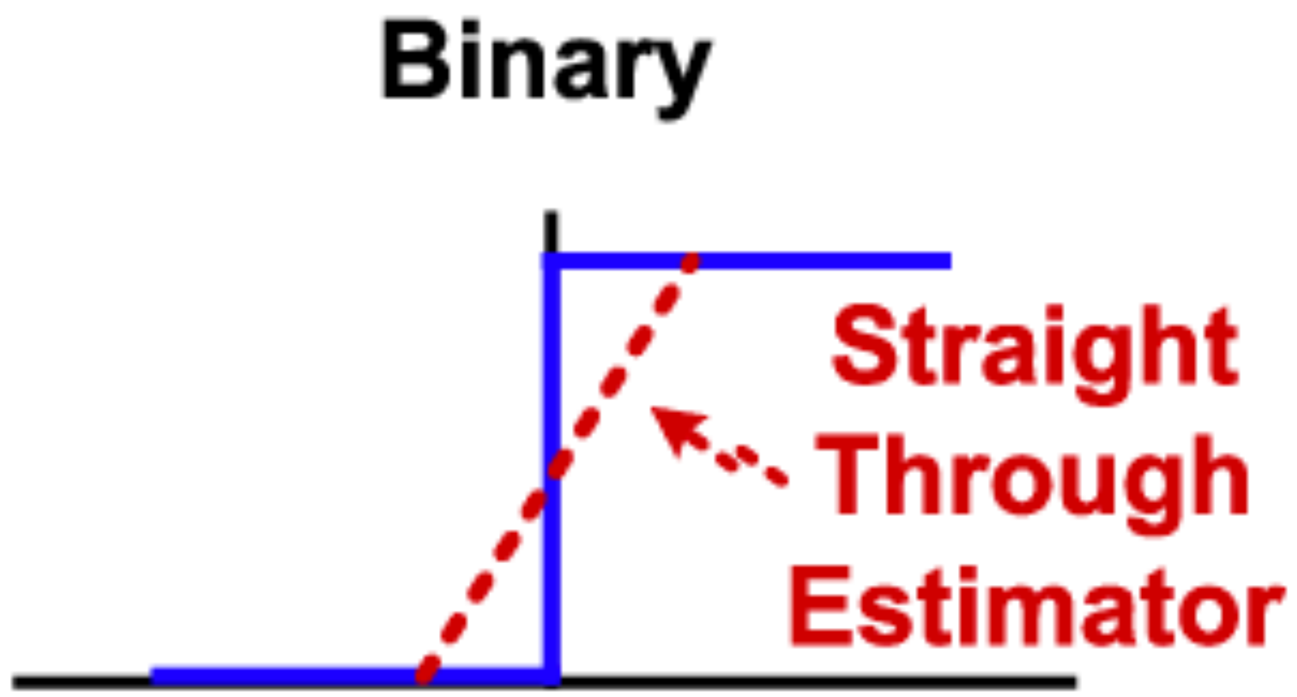
Quantization-aware training: how does it work?

- Fake quantization: using 32-bit floating-point math under the hood
- Straight-through estimator: during backpropagation, ignore quantization operation (treat as identity)

Fake Quantization:



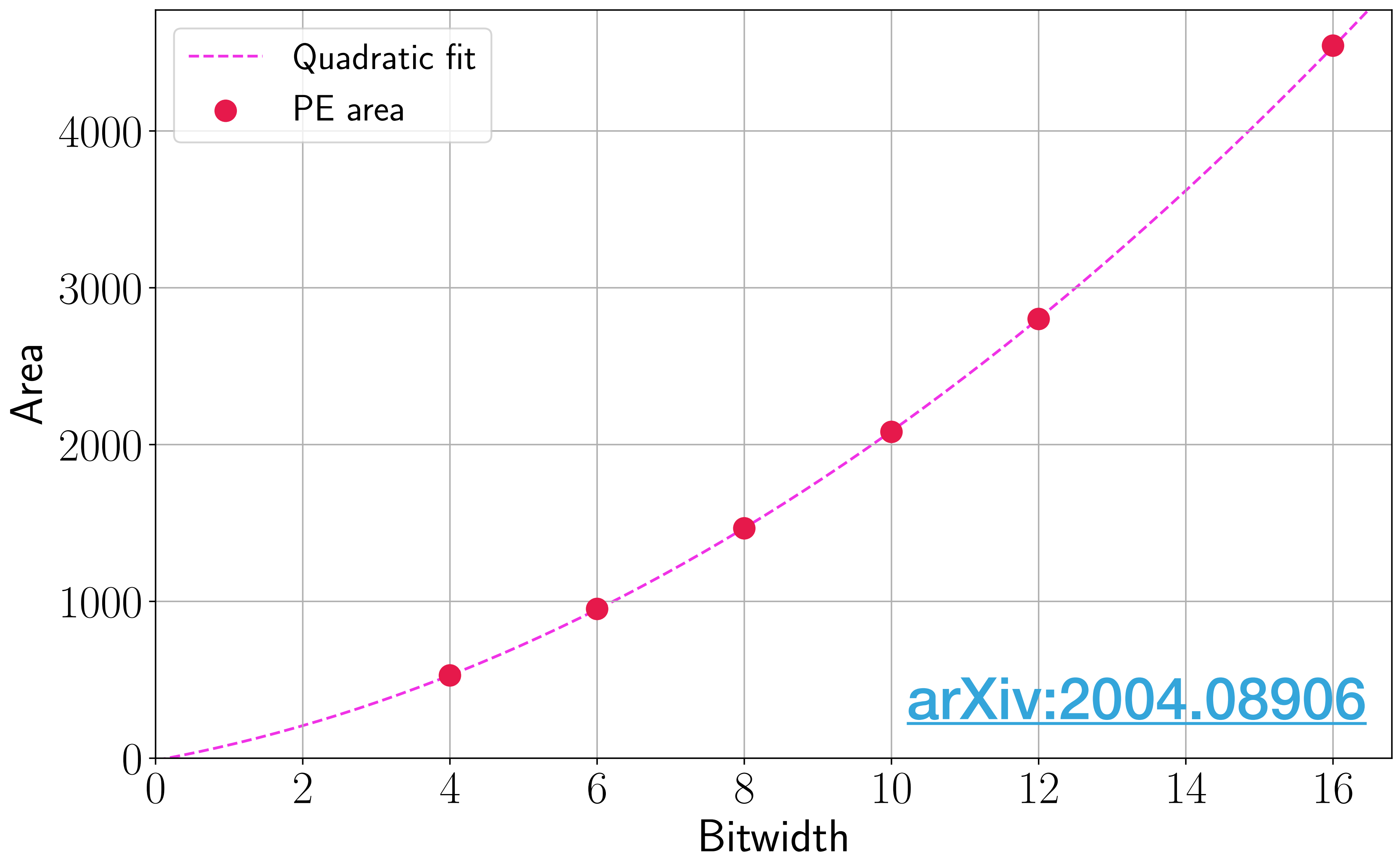


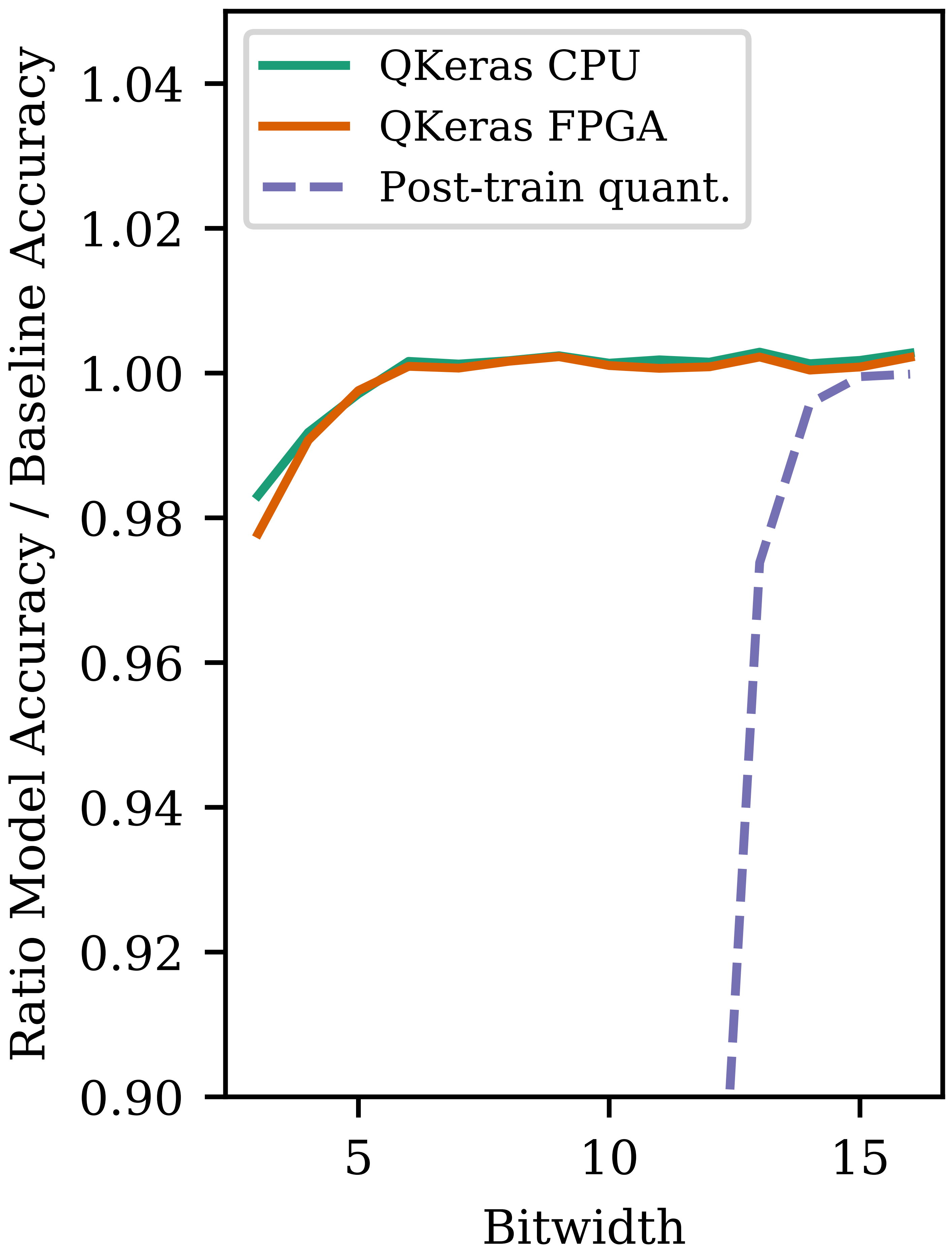


Quantization-aware training

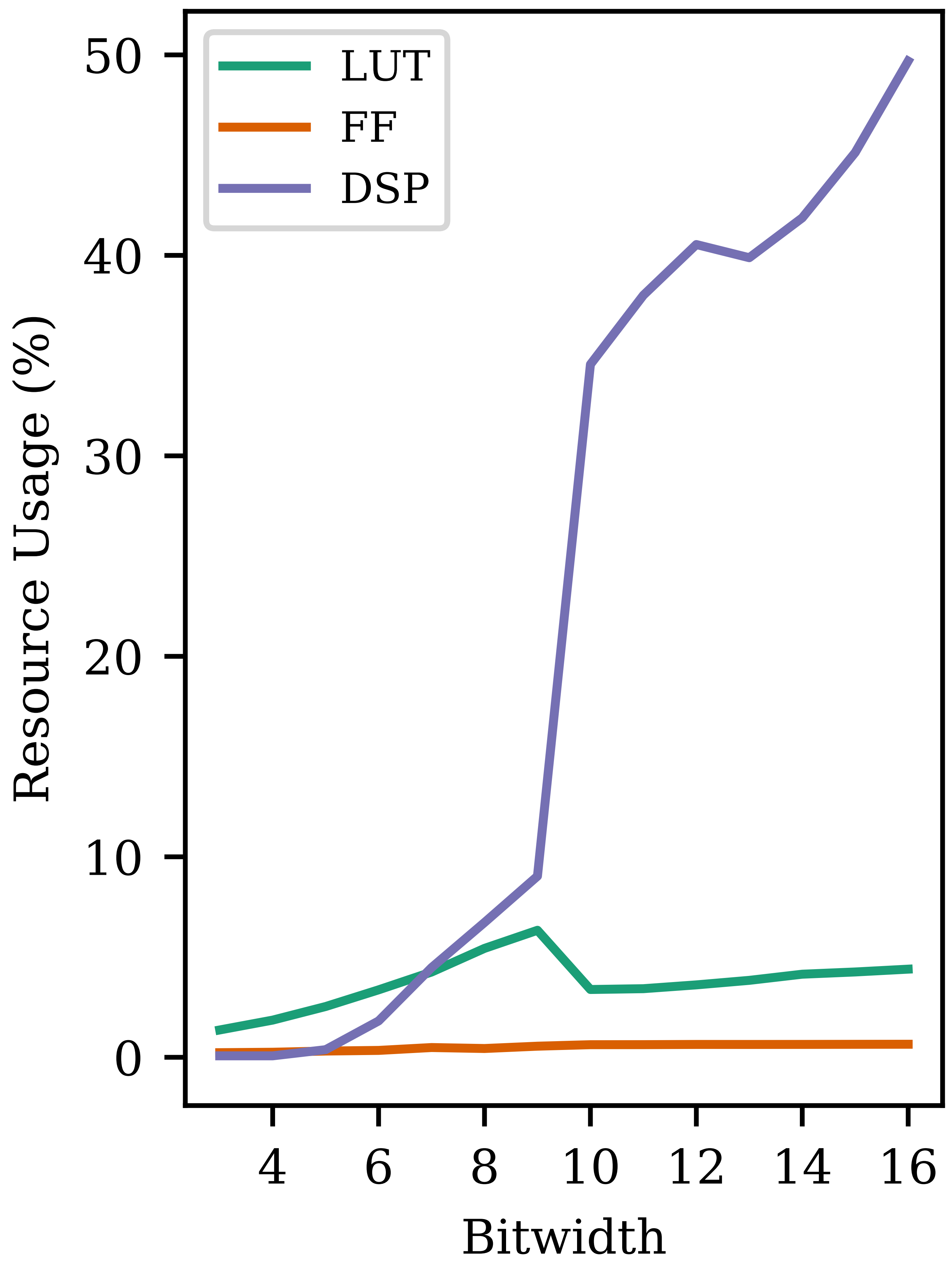
arXiv:2006.10159

- Full performance with 6 bits instead of 14 bits
- Much smaller fraction of resources
- Area & power scale quadratically with bit width





Xilinx VU9P

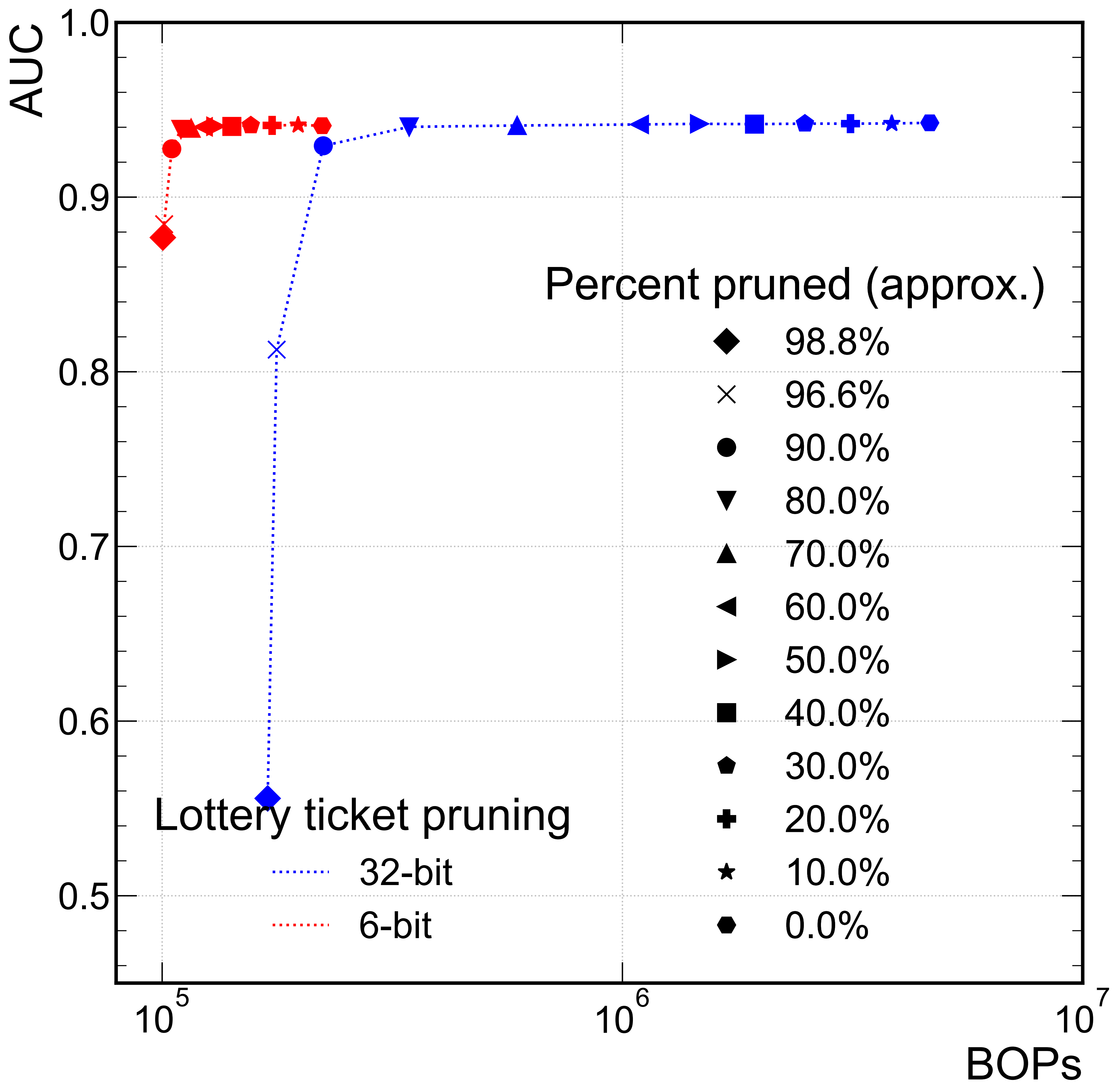


Pruning + quantization-aware training

arXiv:2102.11289

- Quantization-aware pruning (QAP): iterative pruning can further reduce the hardware computational complexity of a quantized model
- After QAP, the 6-bit, 80% pruned model achieves a factor of 50 reduction in BOPs compared to the 32-bit, unpruned model

  - Study using Brevitas

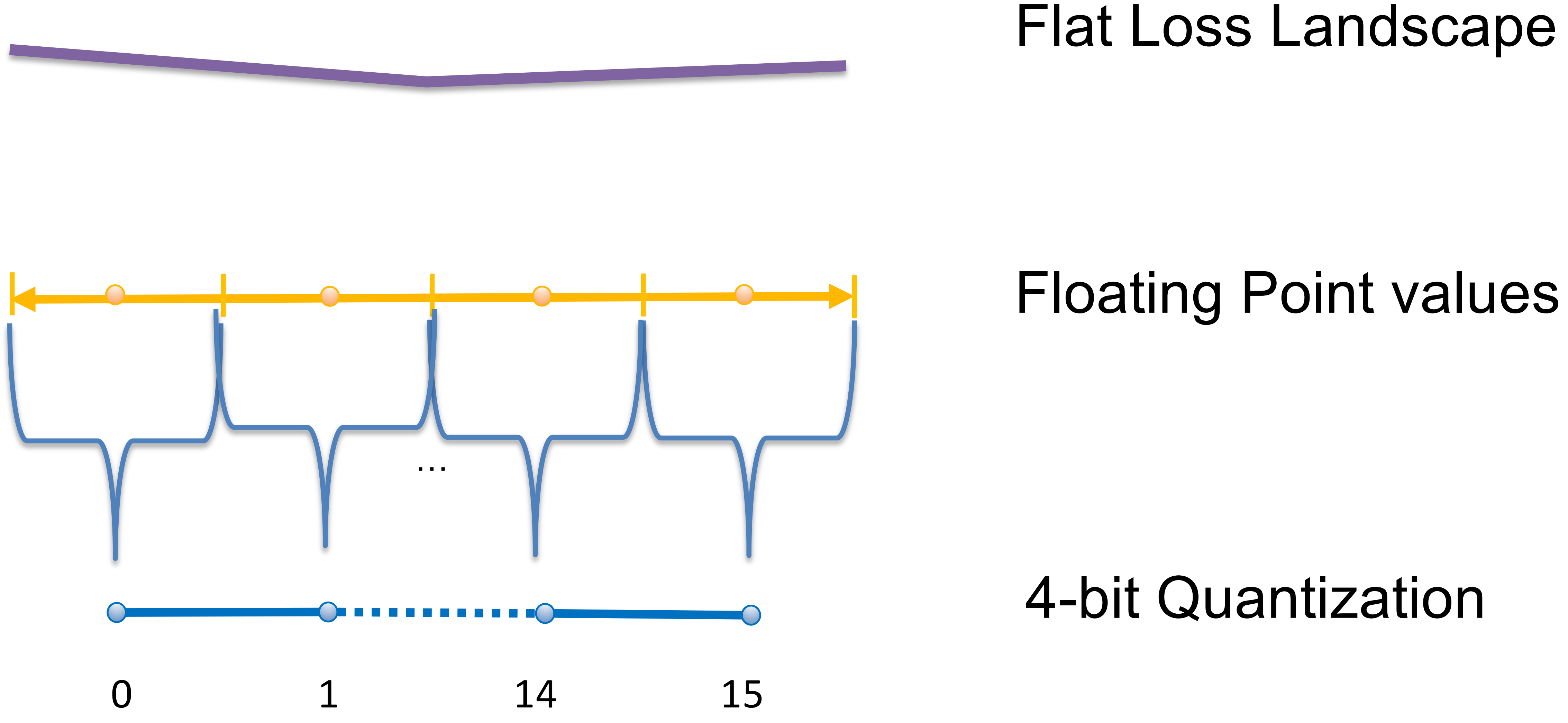


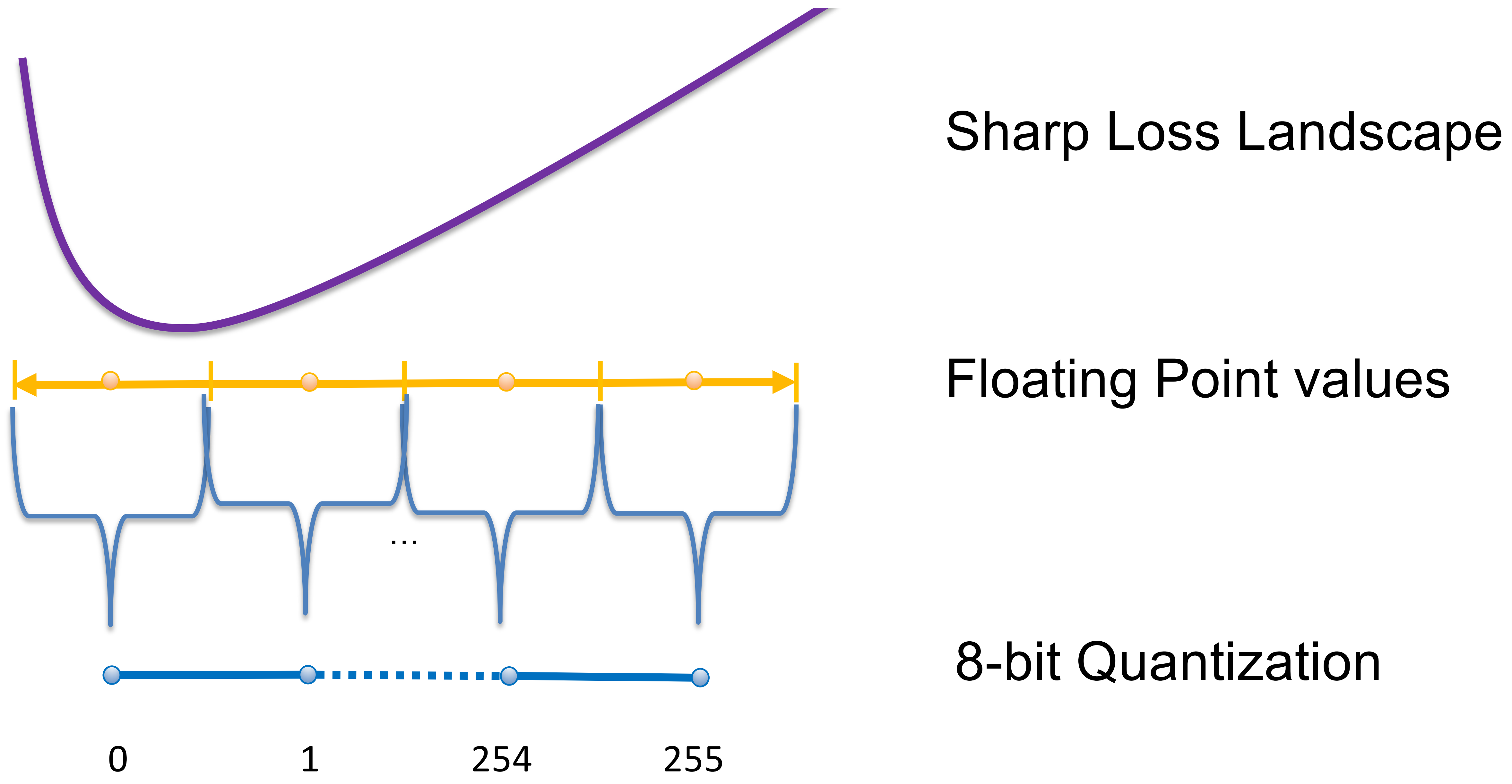
Bit operations (BOPs) definition: arXiv:1804.10969

Hessian-aware quantization (HAWQ)

arXiv:2011.10680

- Hessian of loss can provide additional guidance about quantization!
- Flat loss landscape: Lower bit width
- Sharp loss landscape: Higher bit width





Recap: Pruning and quantization

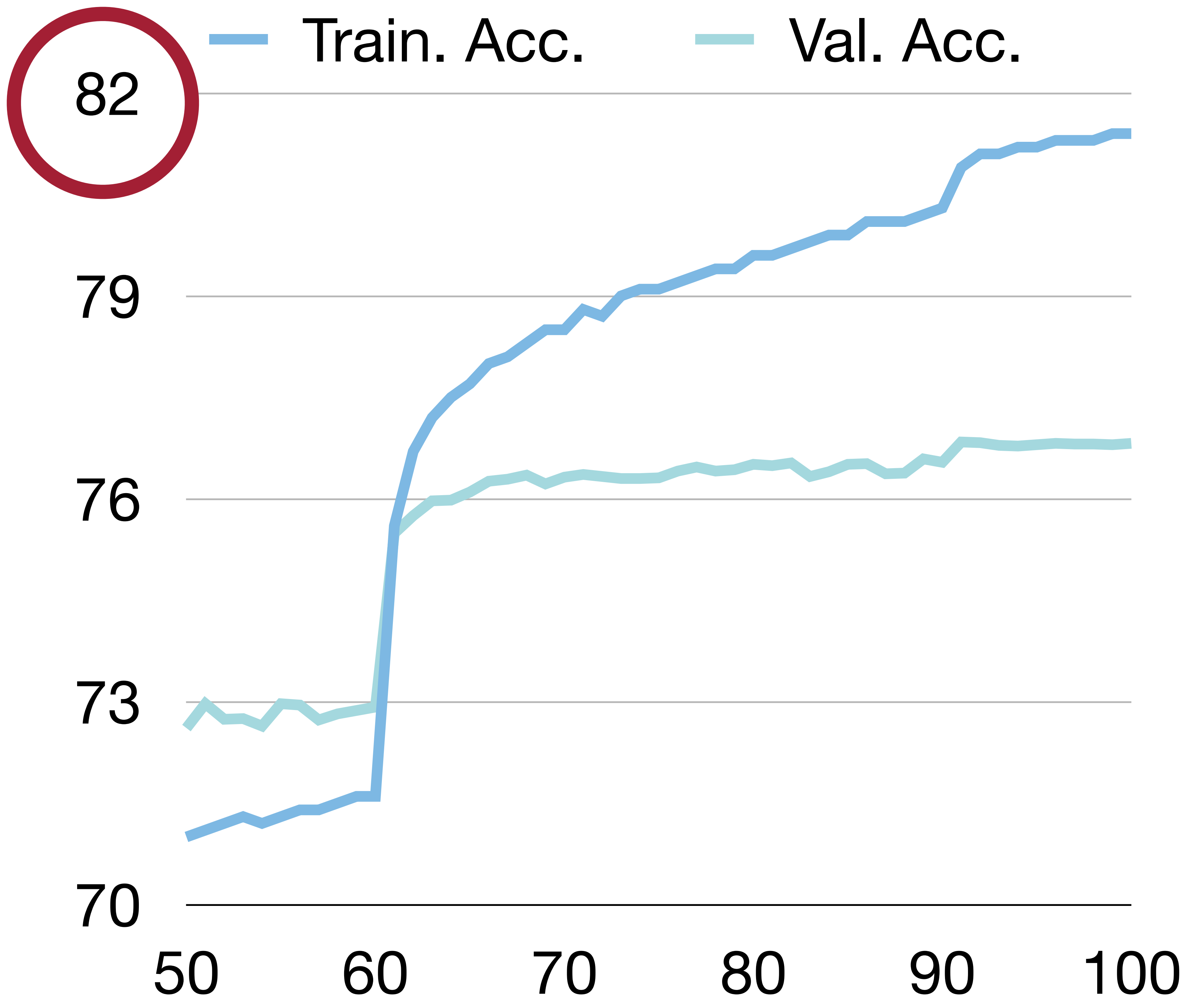
- Pruning and quantization can be used post-training to compress models
- They can also be used more effectively during training to achieve even higher levels of compression

- But so far we haven’t touch the model architecture?
  - Are there compression schemes that do that?
  - Yes, knowledge distillation!

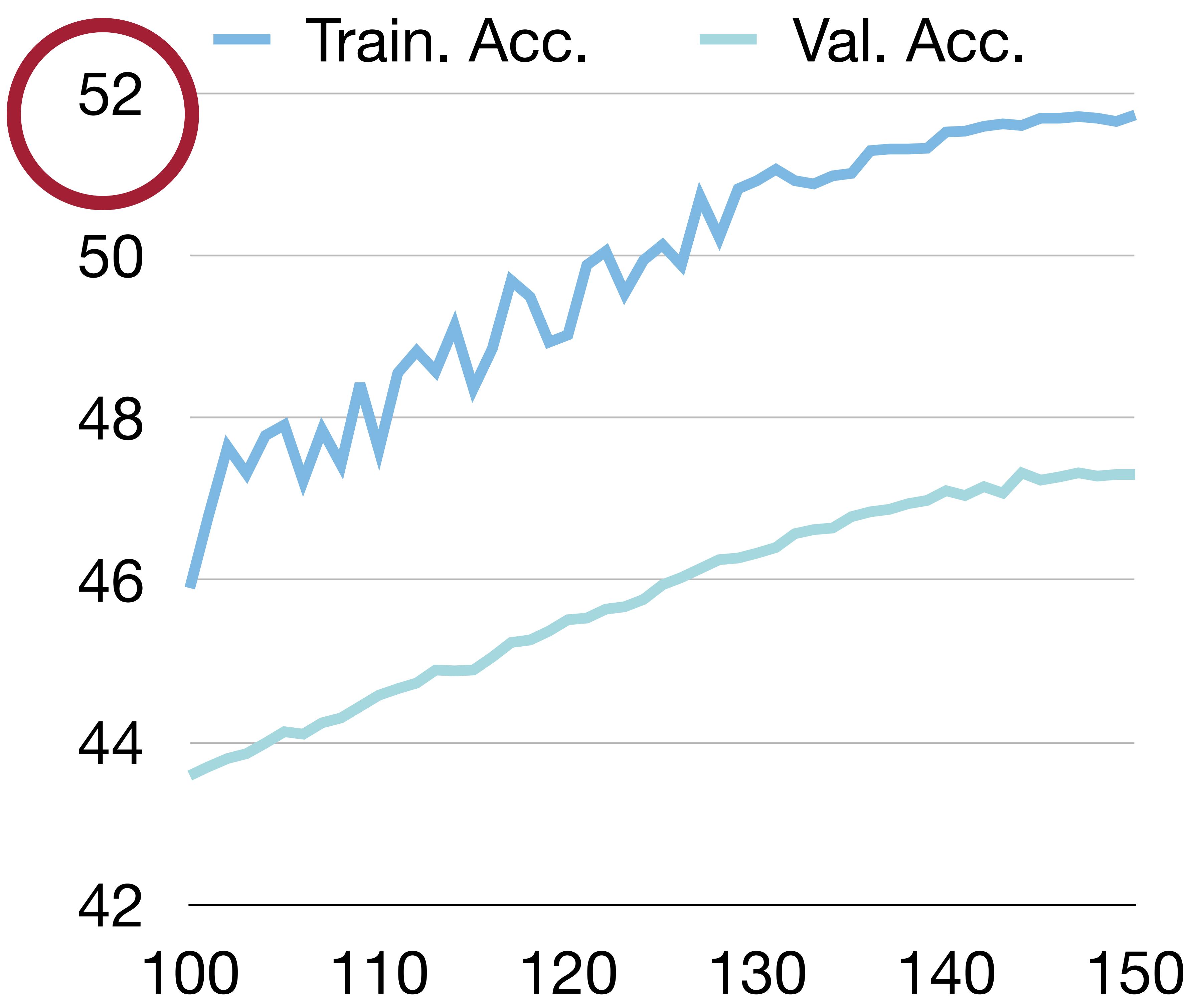
Challenge: tiny models are hard to train

Tiny models underfit large datasets

Training curve for ResNet50



Training curve for MobileNetV2-Tiny



Question: Can we help the training of tiny models with large models?

Knowledge distillation

arXiv:2006.05525

Distilling the Knowledge in a Neural Network

Geoffrey Hinton[*, †]
Google Inc.
Mountain View
geoffhinton@google.com

Oriol Vinyals[†]
Google Inc.
Mountain View
vinyals@google.com

Jeff Dean
Google Inc.
Mountain View
jeff@google.com

Abstract

A very simple way to improve the performance of almost any machine learning algorithm is to train many different models on the same data and then to average their predictions [3]. Unfortunately, making predictions using a whole ensemble of models is cumbersome and may be too computationally expensive to allow deployment to a large number of users, especially if the individual models are large neural nets. Caruana and his collaborators [1] have shown that it is possible to compress the knowledge in an ensemble into a single model which is much easier to deploy and we develop this approach further using a different compression technique. We achieve some surprising results on MNIST and we show that we can significantly improve the acoustic model of a heavily used commercial system by distilling the knowledge in an ensemble of models into a single model. We also introduce a new type of ensemble composed of one or more full models and many specialist models which learn to distinguish fine-grained classes that the full models confuse. Unlike a mixture of experts, these specialist models can be trained rapidly and in parallel.

Illustration of KD

arXiv:2006.05525

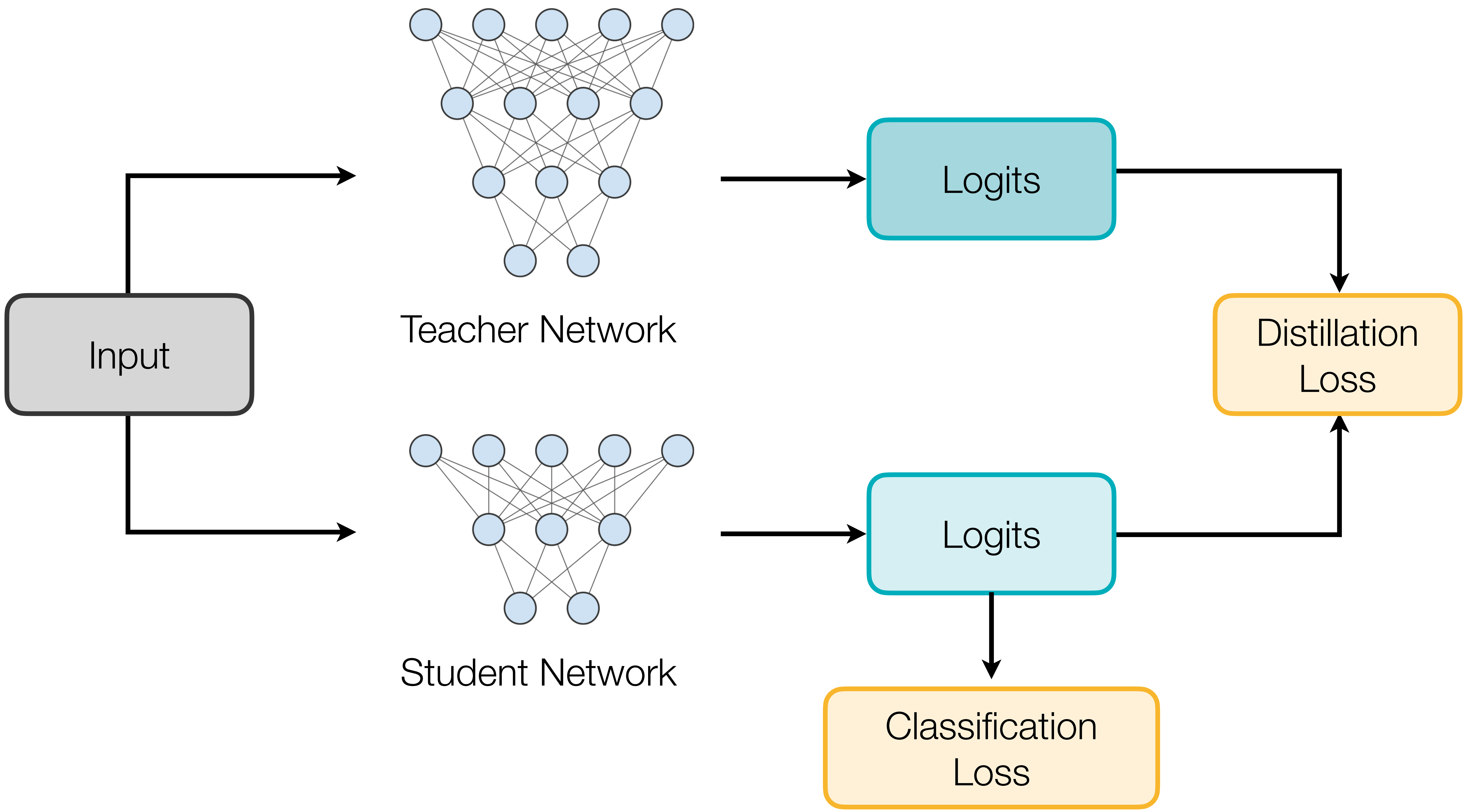
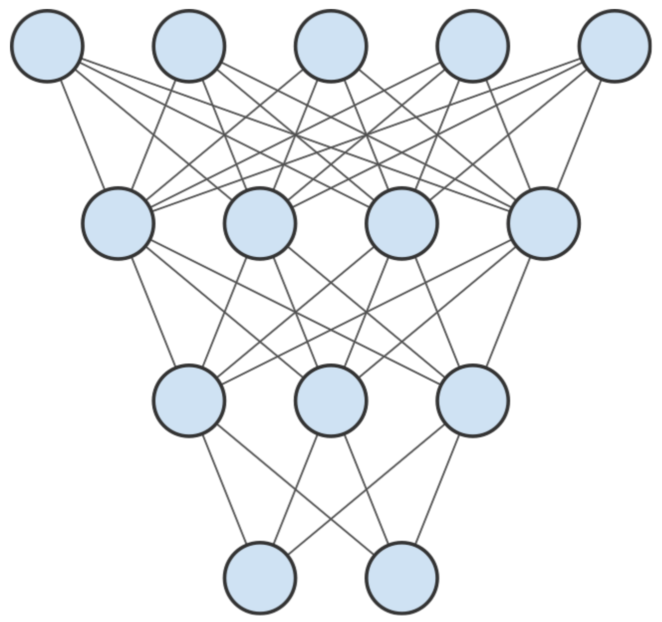


Illustration of KD

Matching prediction probabilities between teacher and student

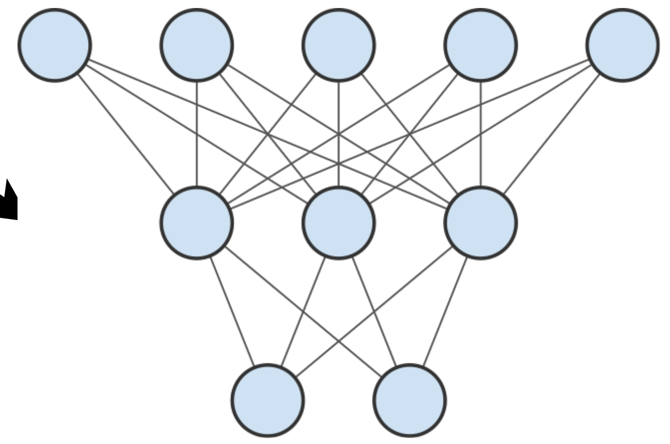


Teacher Network

	Logits	Probabilities
Cat	5	0.982
Dog	1	0.017

$$\frac{\exp(5)}{\exp(5)+\exp(1)}$$

$$\frac{\exp(1)}{\exp(5)+\exp(1)}$$



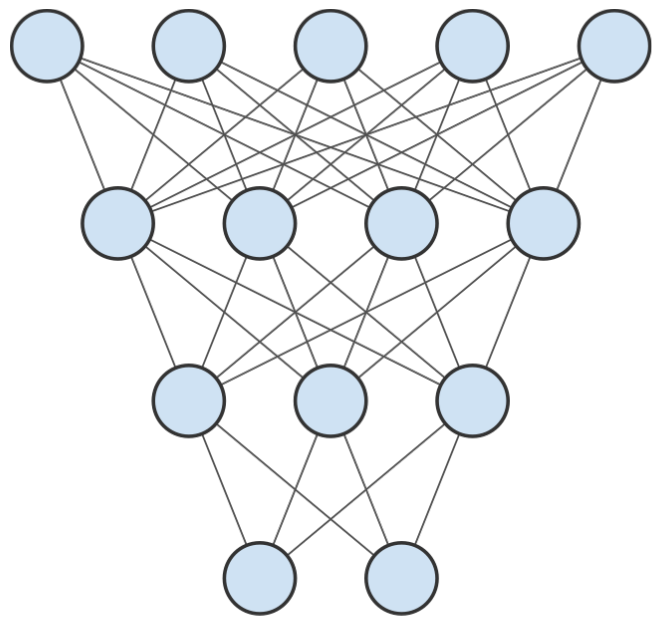
Student Network

	Logits	Probabilities
Cat	3	0.731
Dog	2	0.269

The student model is less confident

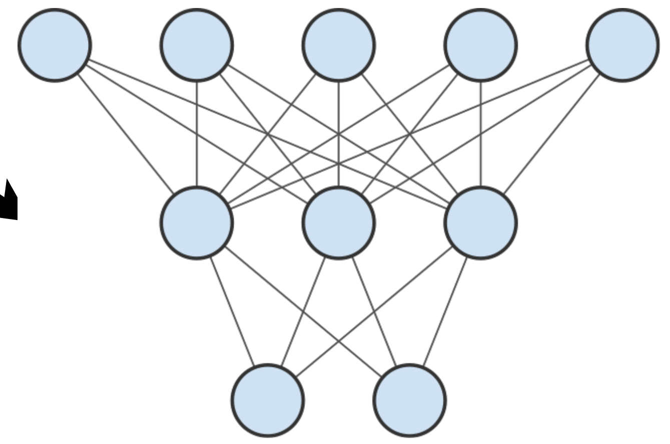
Illustration of KD

Matching prediction probabilities between teacher and student



Teacher Network

	Logits	Probabilities
Cat	5	0.982
Dog	1	0.017

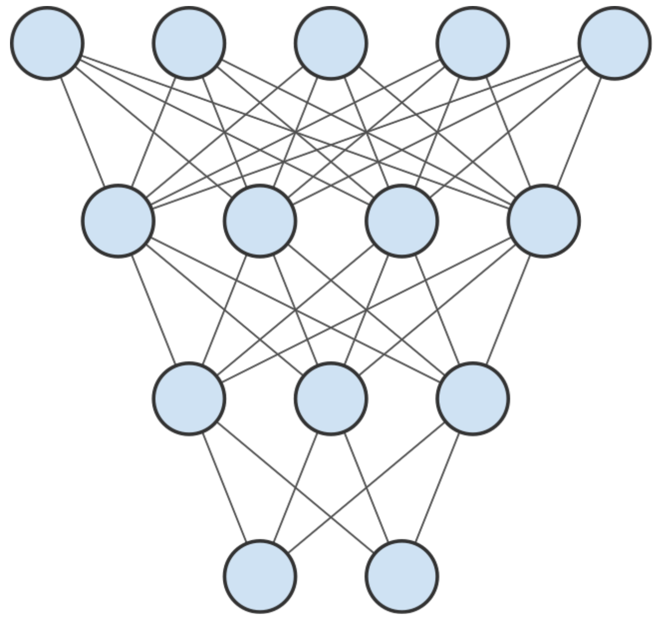


Student Network

	Logits	Probabilities
Cat	3	0.731
Dog	2	0.269

Illustration of KD

Concept of temperature



Teacher Network

$$\frac{\exp(5/1)}{\exp(5/1) + \exp(1/1)}$$

	Logits	Probabilities (T=1)	Probabilities (T=10)
Cat	5	0.982	0.599
Dog	1	0.017	0.401

$$\frac{\exp(5/10)}{\exp(5/10) + \exp(1/10)}$$

A larger temperature smooths the output probability distribution.

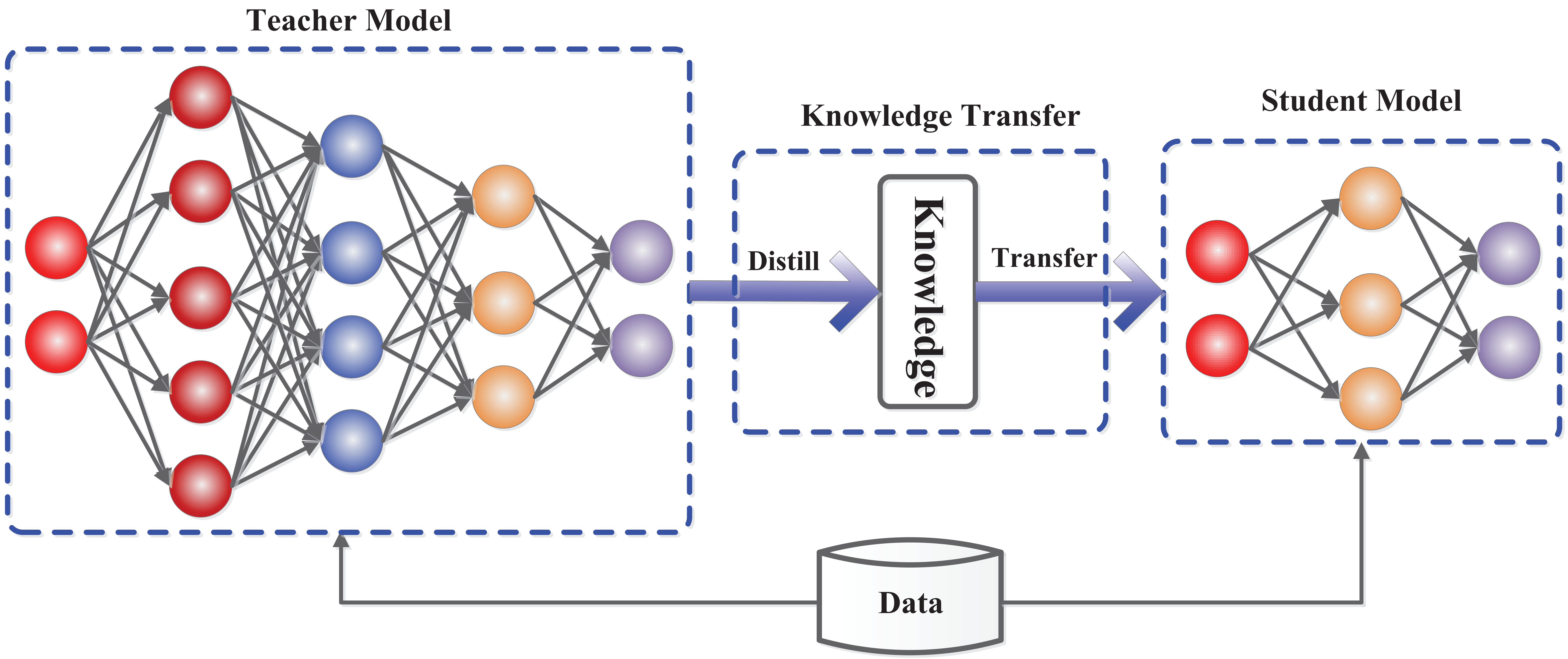
Formal definition of KD

- Neural networks typically use a softmax function to generate the **logits** z_i to class **probabilities** $p(z_i, T) = \frac{\text{exp}(z_i/T)}{\sum_j \text{exp}(z_j/T)}$. Here,$i, j = 0,1,2,...,C - 1$, where C is the number of classes. T is the temperature, which is normally set to 1.

- The goal of knowledge distillation is to **align the class probability distributions from teacher and student networks**.

KD summary

arXiv:2006.05525



- Knowledge distillation: training a small student network to emulate a larger teacher model or ensemble of networks

$$\mathcal{L}_s := \alpha \mathcal{L}_{\mathrm{NLL}} + (1-\alpha)\mathcal{L}_{\mathrm{KD}}$$

$$\mathcal{L}_{\mathrm{NLL}}(\mathbf{z}_s, \mathbf{y}) := -\sum_{j=1}^{c} y_j \log \sigma_j(\mathbf{z}_s), \quad \mathcal{L}_{\mathrm{KD}}(\mathbf{z}_s, \mathbf{z}_t) := -\tau^2 \sum_{j=1}^{c} \sigma_j\left(\frac{\mathbf{z}_t}{\tau}\right) \log \sigma_j\left(\frac{\mathbf{z}_s}{\tau}\right)$$

Next time

- Guest lecture!