

CNN.tar.gz

Compression for fast CNNs with FPGA

Luigi Sabetta - Offshell 2021

Stefano Giagu, Graziella Russo, Simone Francescato,
Federtica Riti, Federico Tortonesi

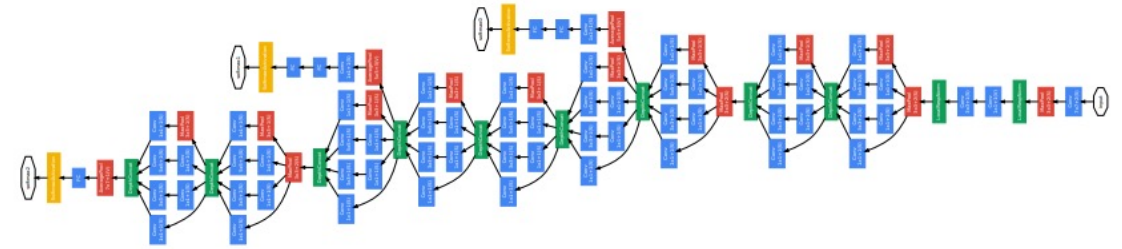


SAPIENZA
UNIVERSITÀ DI ROMA



The Problem

- Nowadays **ML algorithms** are getting **bigger and bigger** in size to reach higher and higher performance



GoogleNet Latency: $\sim ms$



- In many application **low latency** and **minimal resources-utilization/energy consumption** are although the main limitation

The solution

Compression and simplification!

We present the combined result of 3 general use and effective compression techniques:

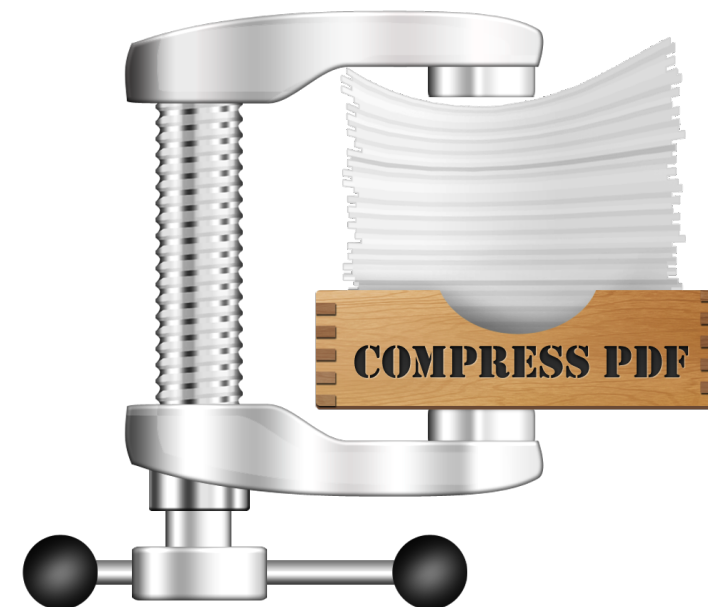
- Input Fragmentation



- Knowledge Distillation



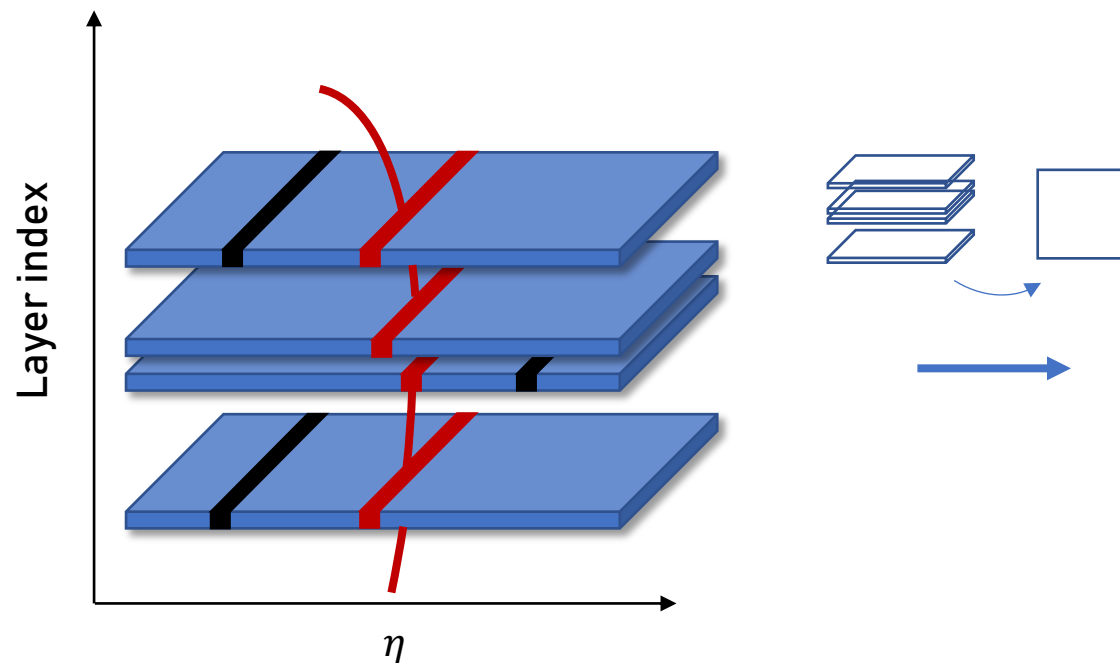
- Quantization



DataSample/structure

Realistic toy simulation of a HEP muon detector (RPC) of the ATLAS experiment

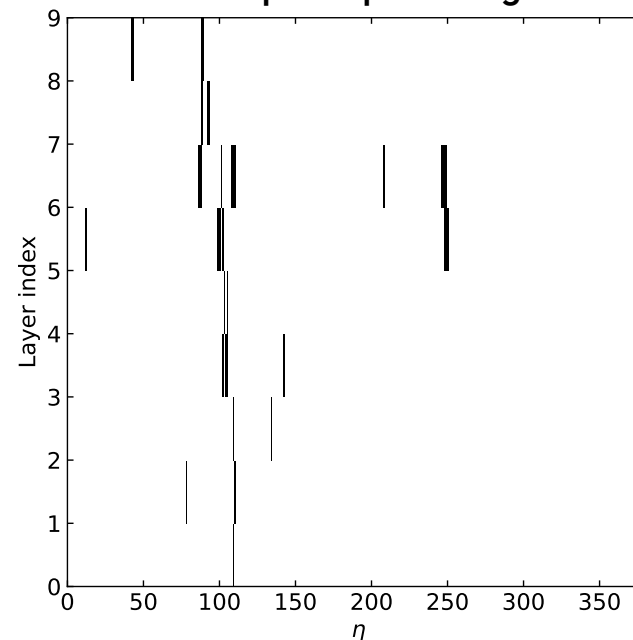
- Track bended due to magnetic field
- Electronic + experimental noise added



- 700k images

Target: (p_T, η)
 $3 < p_T < 20 \text{ GeV}$

Example input image



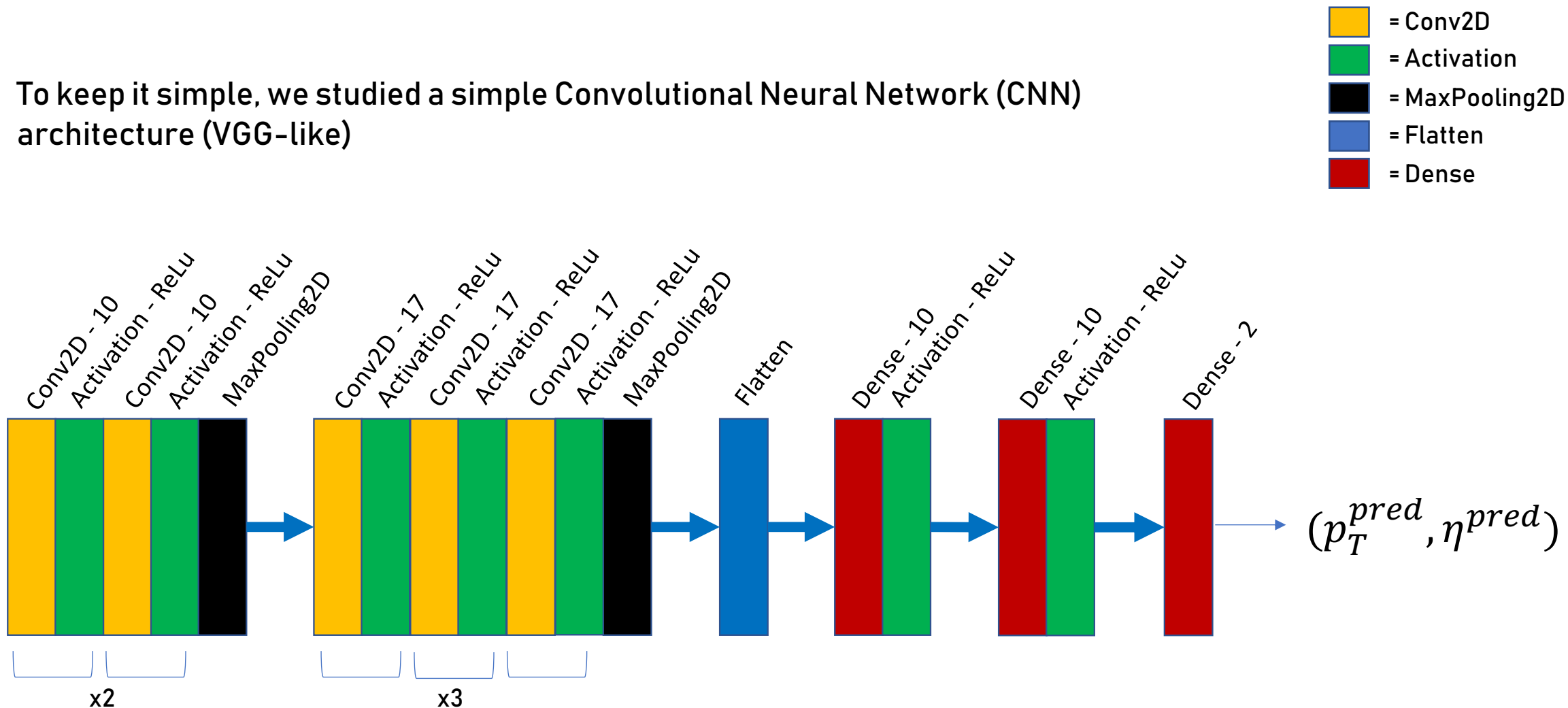
Constraints

Three main aspects guided the choices taken for this project:

- **Occupancy**: fit within the FPGA resources
- **Latency**: run in less than ~ 400 ns.
- **Fake Rate**: less than $\sim 2\%$.

Teacher architecture

To keep it simple, we studied a simple Convolutional Neural Network (CNN) architecture (VGG-like)



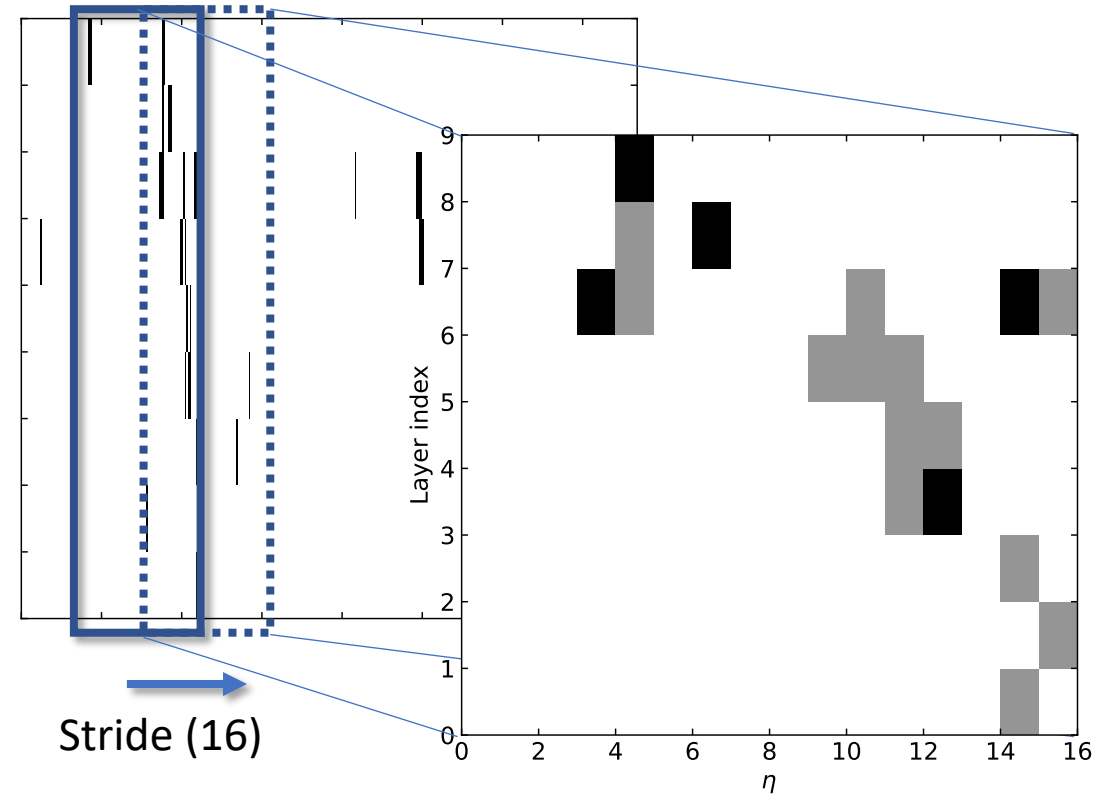
Input fragmentation

9x384
Pixels

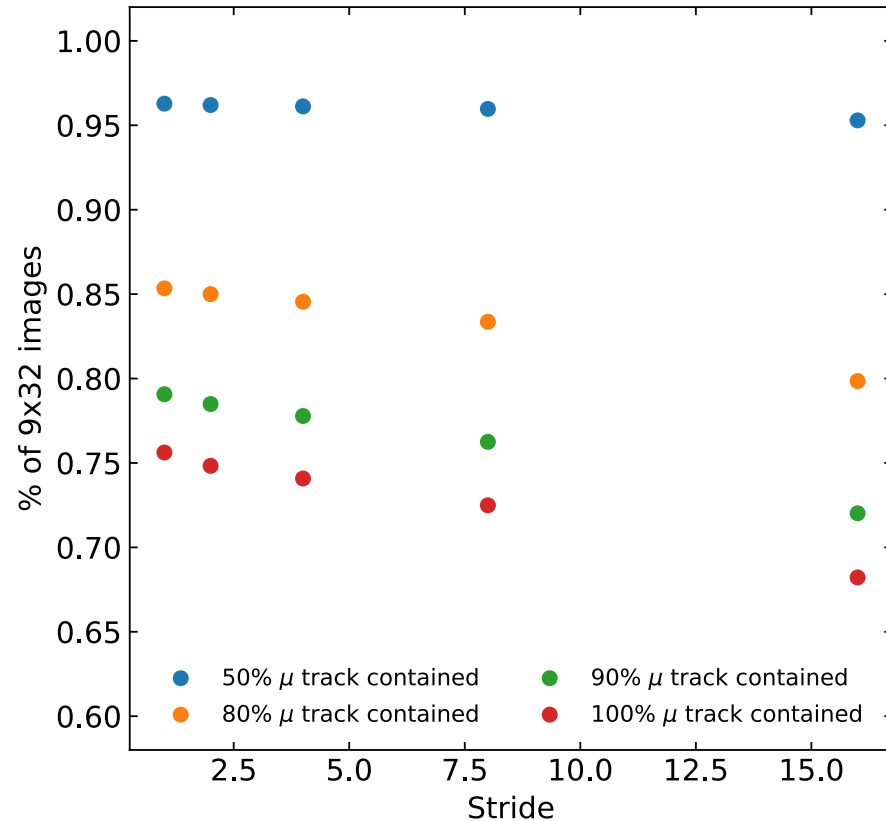
24 times smaller input

9x16
Pixels

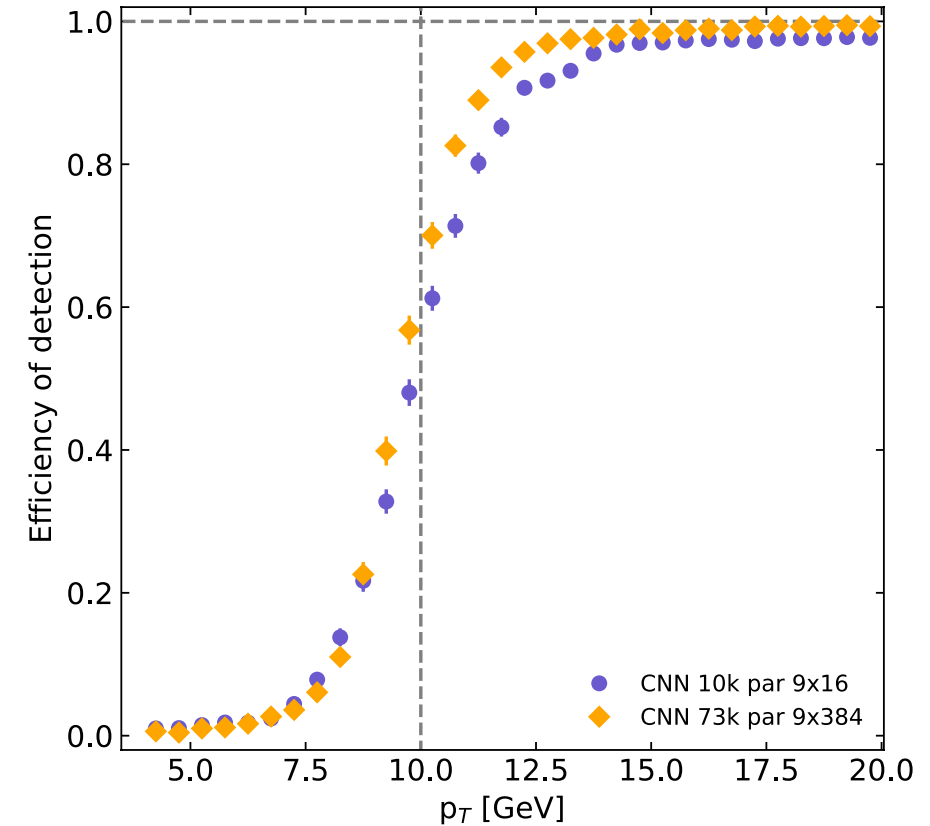
- Slide a 9x32 sector with variable stride
- Select the 9x32 sector with the largest number of hits
- AveragePool (1,2) to halve the number of pixels



Input fragmentation

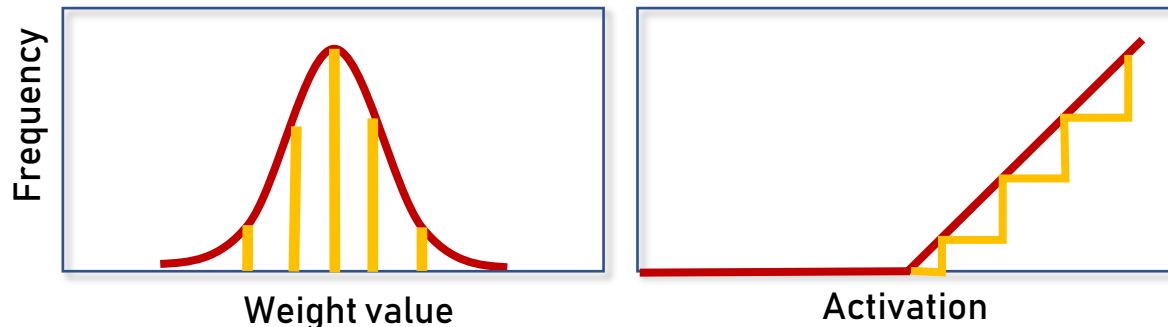


- 50% of the particle track is contained in more than 90% of the fragments regardless of the stride



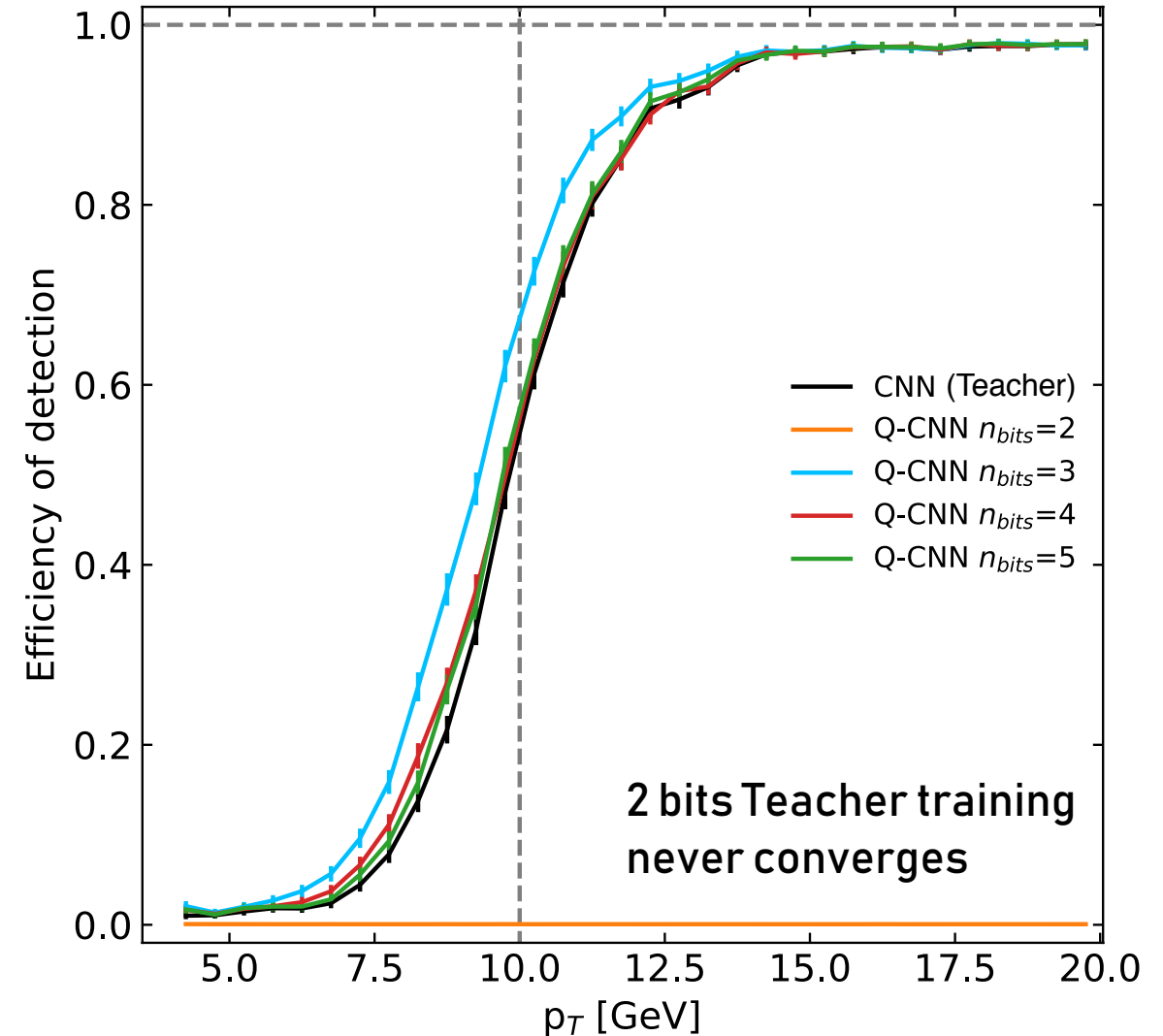
Quantization

- Each weight of the network can be described with diminished precision

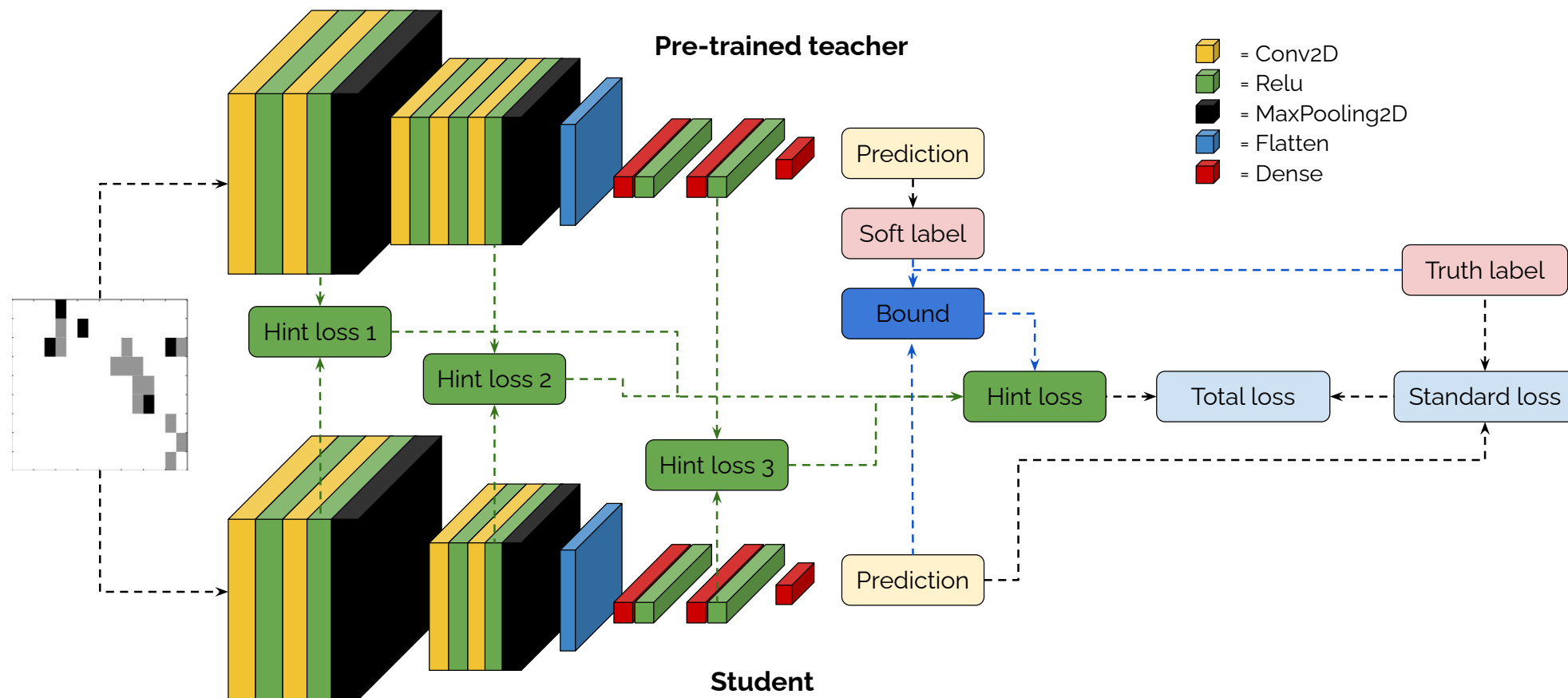


Quantization-Aware Training

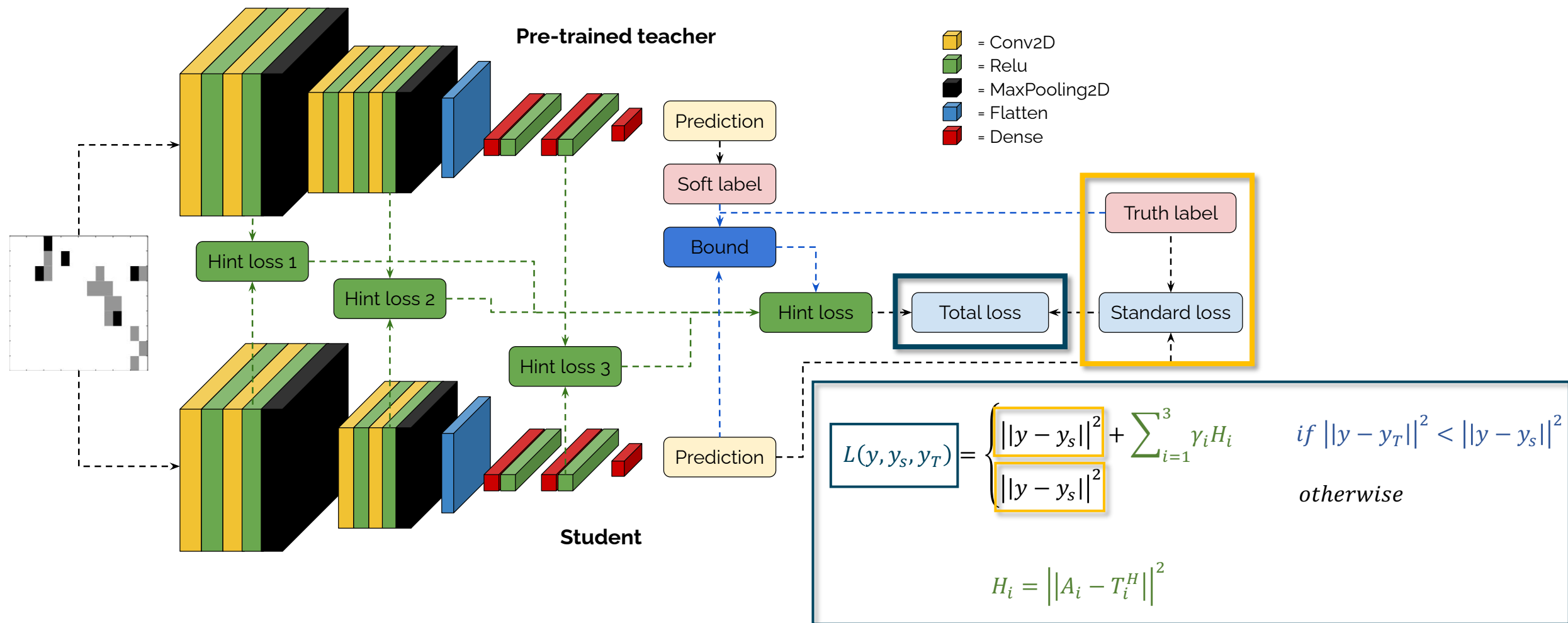
We quantized uniformly **every part** of the network but the last layer **BEFORE** training



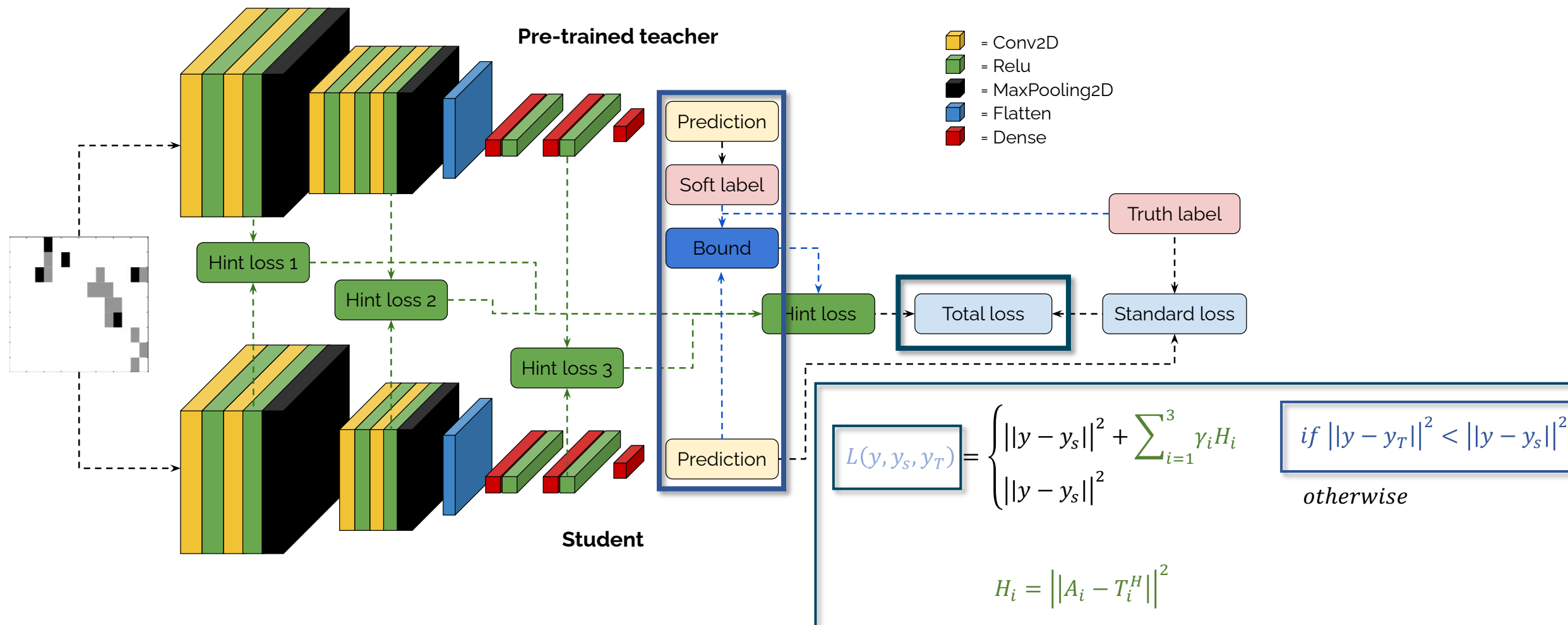
Knowledge Distillation



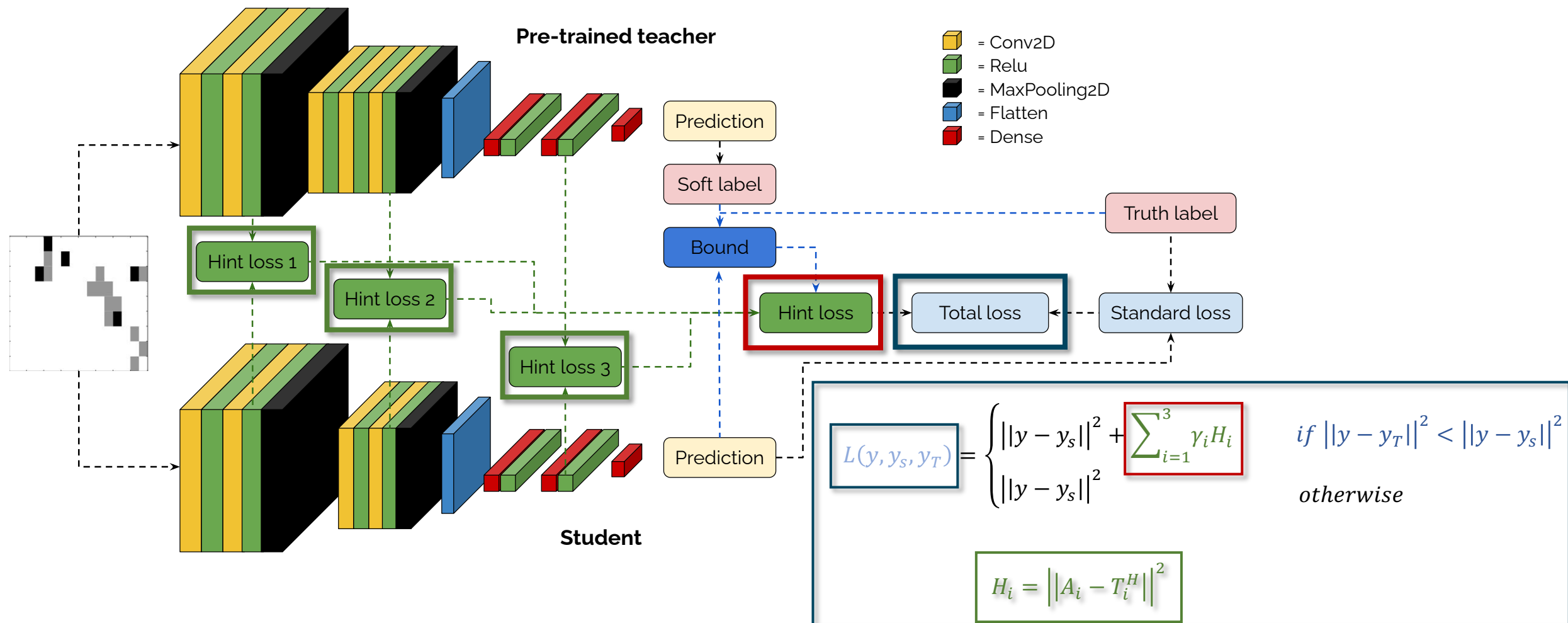
Knowledge Distillation



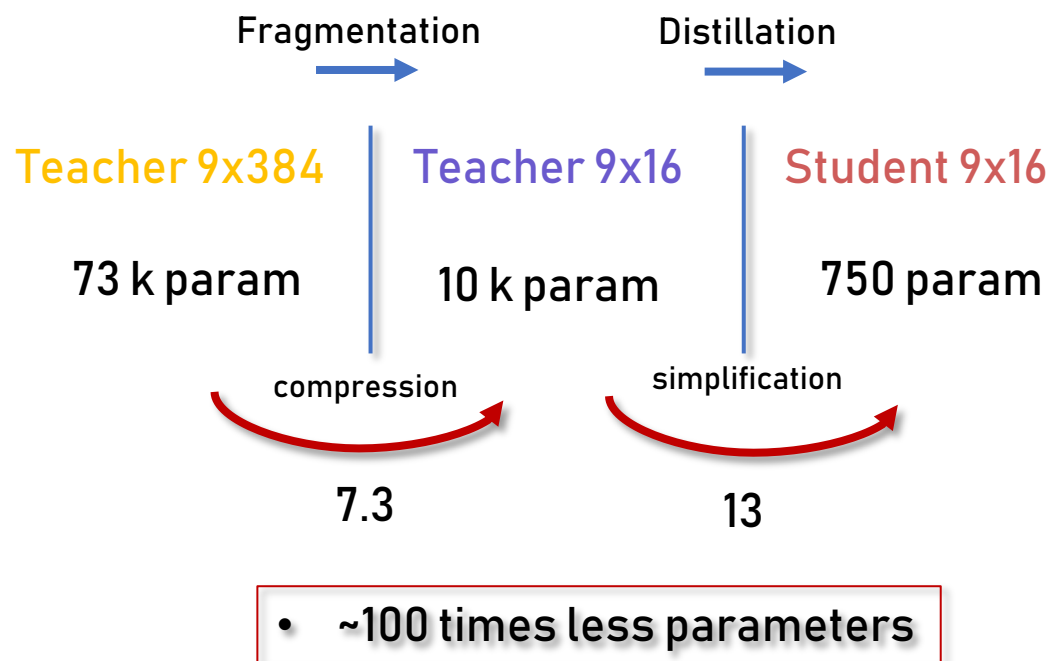
Knowledge Distillation



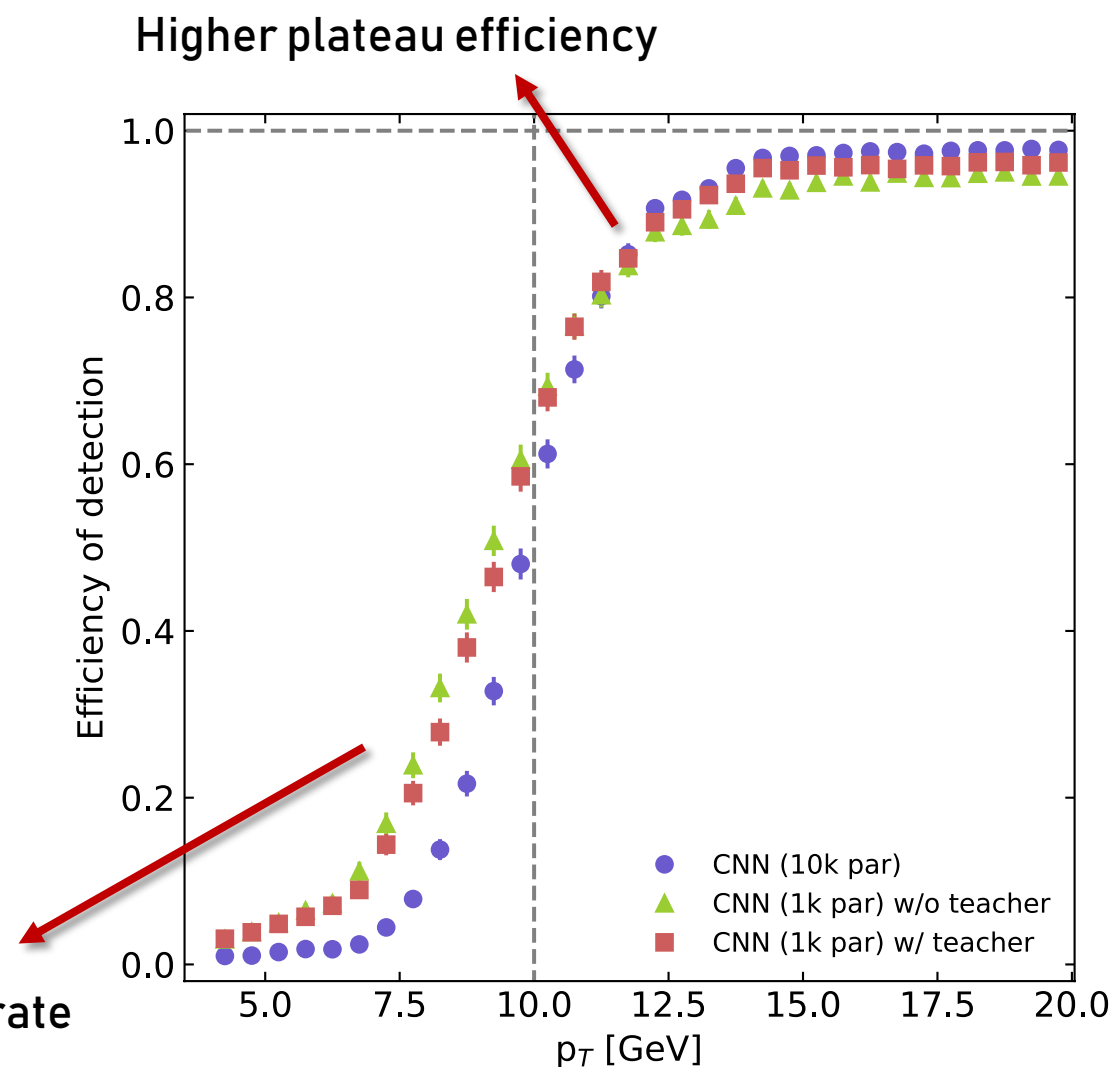
Knowledge Distillation



Knowledge Distillation



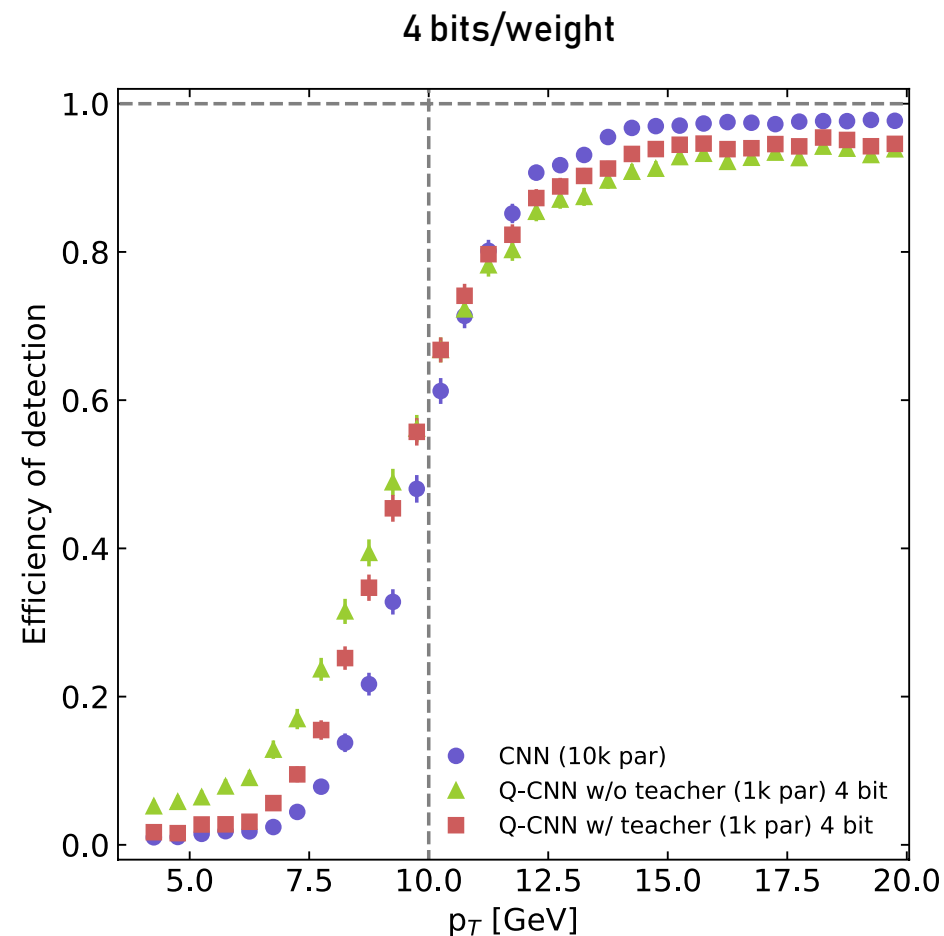
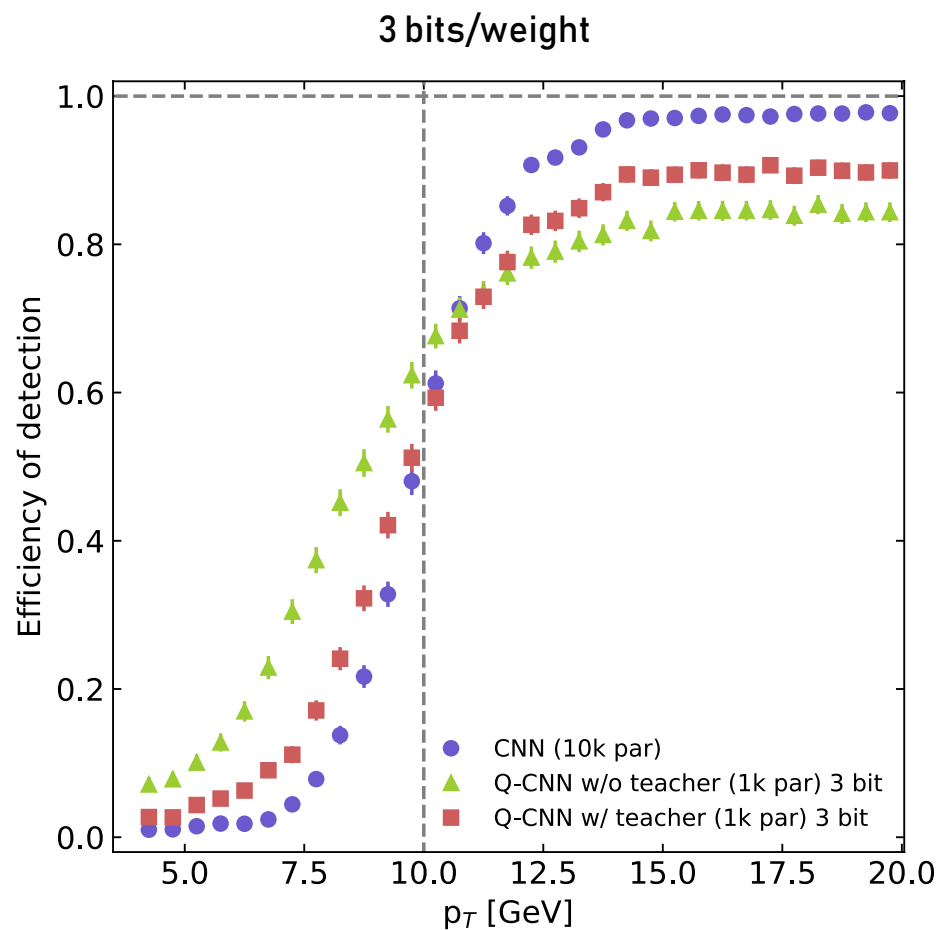
- Modest but evident improvement for 32 bits fp weights



Performance - 1

Efficiency curves

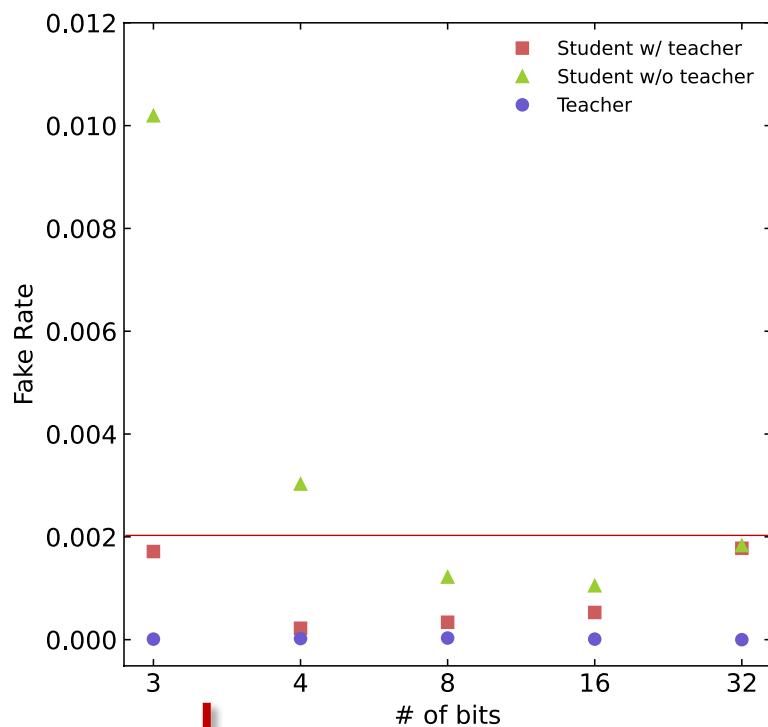
- Even greater improvements from QAT and KD combination



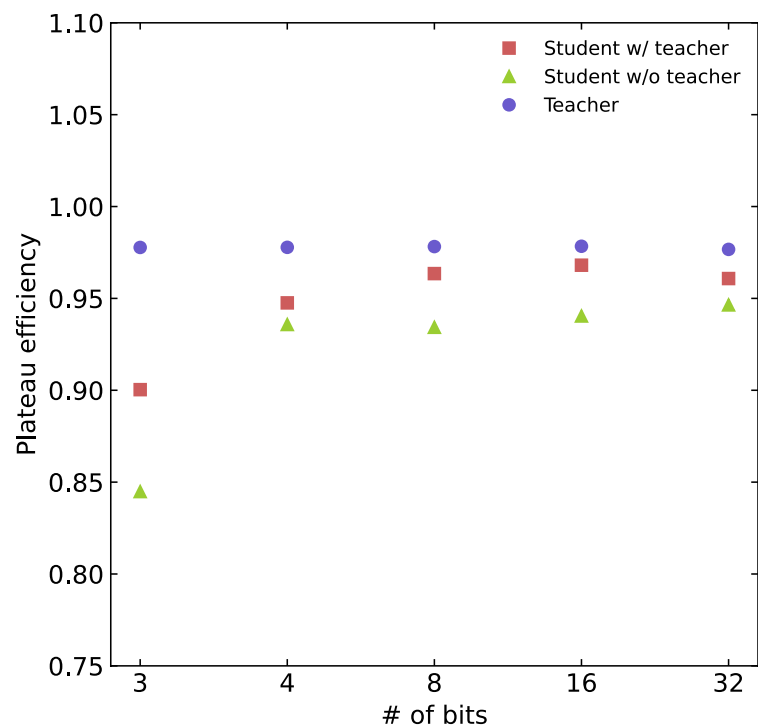
Performance - 2

Physical quantities

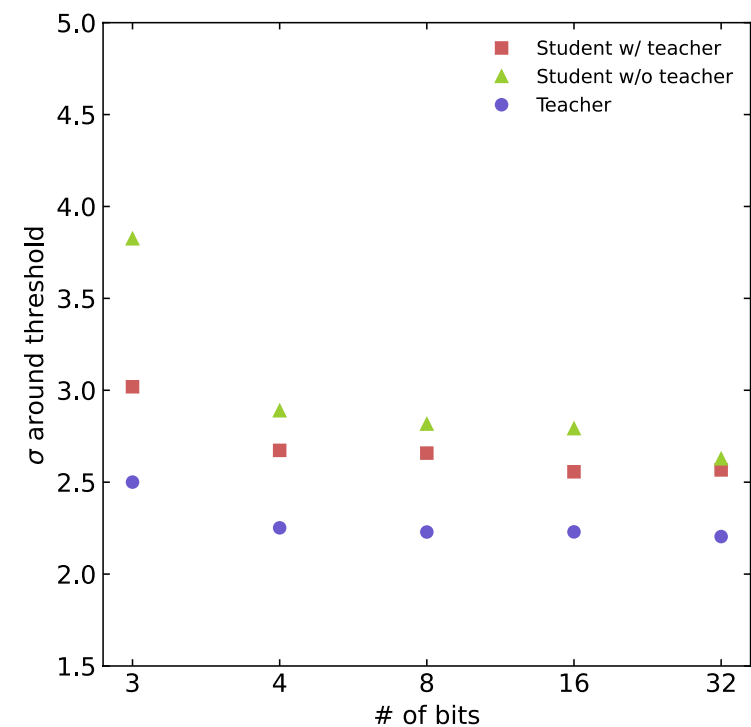
Evident benefits from KD in all the physical metrics



Fake rate constraint reached for all # bits through KD



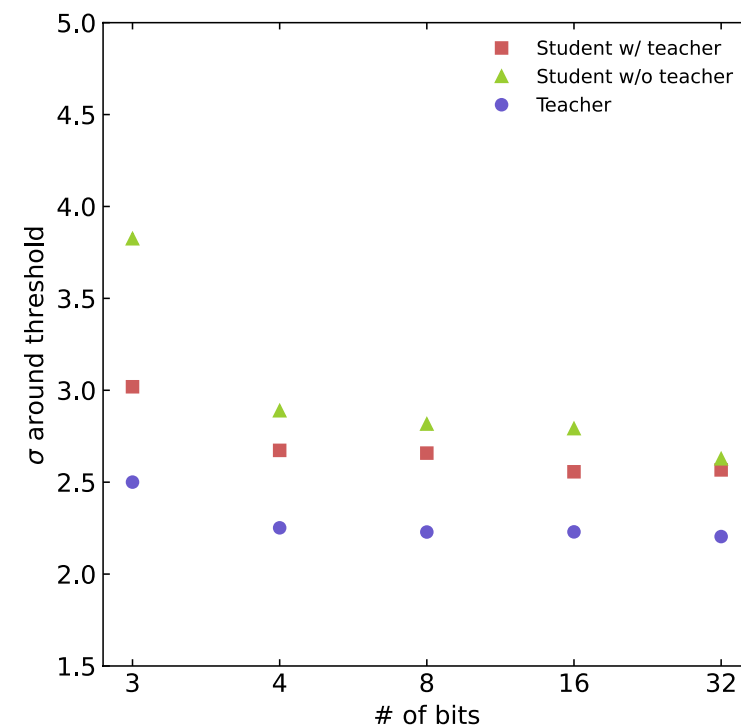
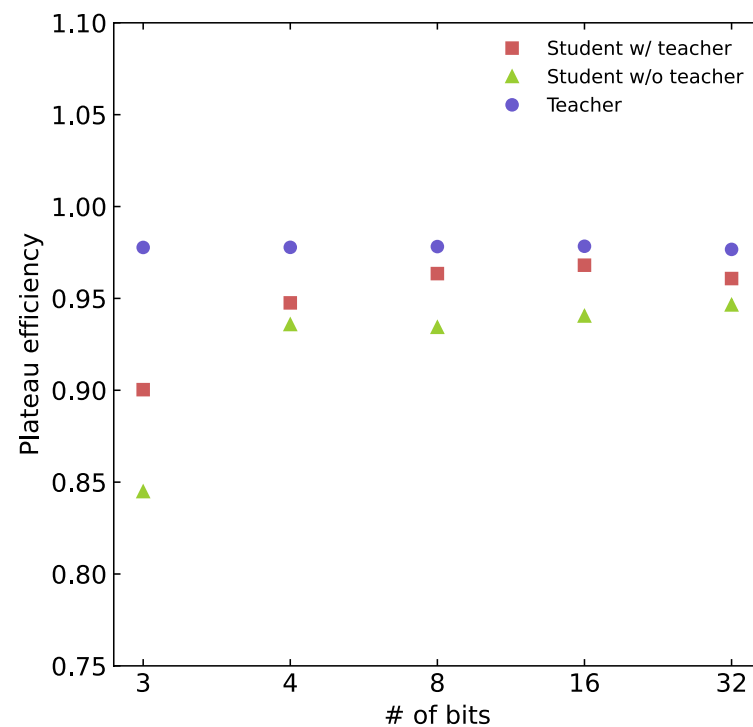
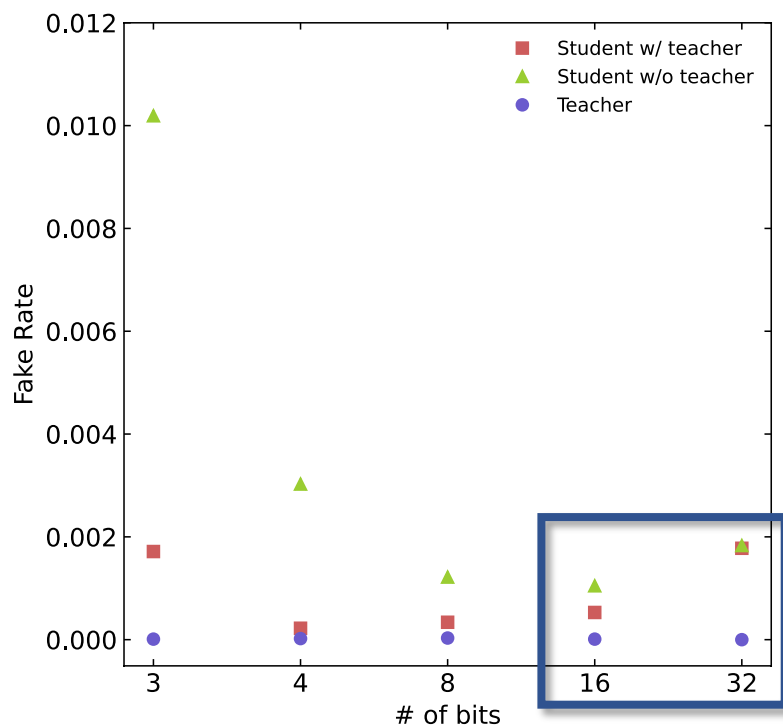
$p_T > 17 \text{ GeV}$



$7.5 < p_T < 12.5 \text{ GeV}$

Performance - 2

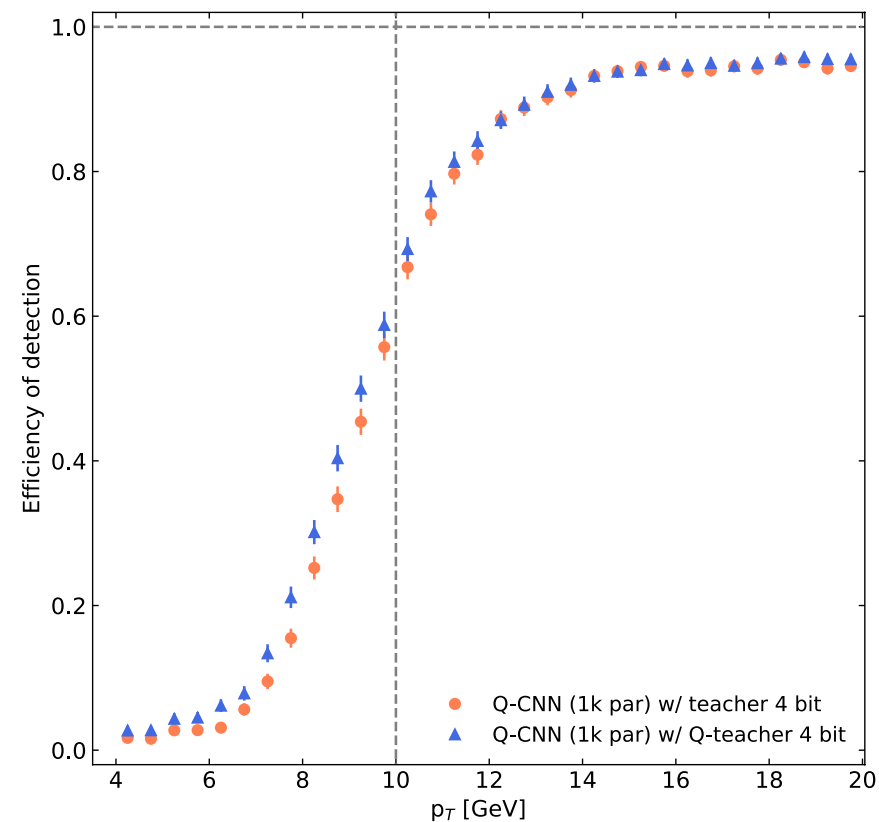
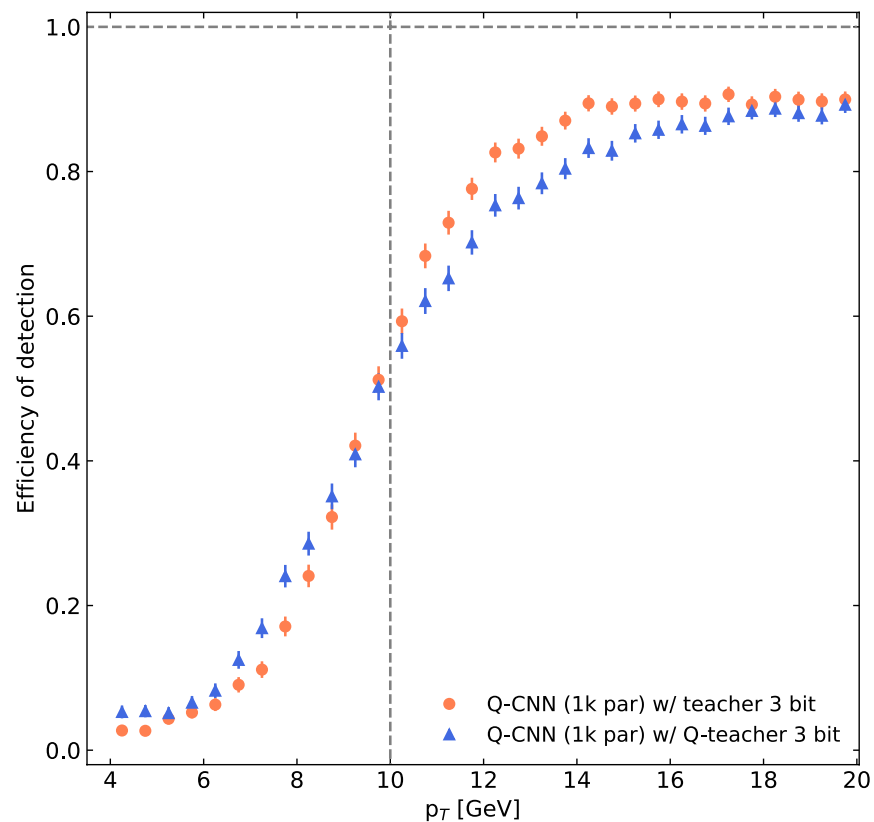
Physical quantities



- Possible explanation: network with higher precision are more likely to reconstruct partial patterns

Performance - 3

Quantized Teacher vs non-quantized Teacher



- A quantized Teacher seems to lead to worse results

Implementation



Model architecture



C++ configuration



VHDL firmware

Resources occupation

Numbers for 9x384 model not reported since not synthesizable

Model (9 × 16)	BRAM	DSPs	FF	LUT	Latency (cycles)
Teacher	1123	31.7 k	2.4 M	265.6 k	640
Student 32 bit	171	3.8 k	247 k	31 k	222
QStudent 4 bit	11	6	14.3 k	29.5 k	183
QStudent 3 bit	11	0	11.6 k	23.3 k	182

- **Occupation** almost **negligible** in respect to total FPGA resources! (Virtex Ultrascale+ 13p)

- **Latency requirement met** for Student models with less than 4 bits/weight (clock period: 2 ns)

Model (9 × 16)	BRAM	DSPs	FF	LUT
Teacher (%)	20	258	69	15
Student 32 bit (%)	3	31	7	1
QStudent 4 bit (%)	~0	~0	~0	1
QStudent 3 bit (%)	~0	~0	~0	1

- **Compression factors relative to Teacher model**

Model (9 × 16)	BRAM	DSPs	FF	LUT	Latency (cycles)
Student 32 bit	6.6	8.3	9.7	8.5	2.9
QStudent 4 bit	102	5.28 k	168	9	3.5
QStudent 3 bit	102	nd	218	11.4	3.5

Conclusions

- We showed an **effective and tunable approach** to reach impressive memory/latency constraint
 - ~ 100 times less weights
 - Latency < 390 ns
 - Fake rate lower than 2%
- We observed a noticeable improvement from the combination of **Fragmentation**, **QAT**, and **KD**