Jet Single Shot Detection

Presented at vCHEP 2021: Artificial Intelligence Session

https://arxiv.org/abs/**2105.05785**

Adrian Alan Pol [1], Thea Aarrestad [1], Katya Govorkova [1], Roi Halily [4], Tal Kopetz [4], Anat Klempner [4], Vladimir Loncar [1,3], Jennifer Ngadiuba [2], Maurizio Pierini [1], Olya Sirkin [4], Sioni Summers [1]

[1] European Organization for Nuclear Research (CERN), Geneva, Switzerland
[2] California Institute of Technology, Pasadena, USA
[3] Institute of Physics Belgrade, Belgrade, Serbia
[4] CEVA, Herzliya, Israel

email: adrianalan.pol@cern.ch

20th May 2021

Motivation

- Looking for novel physics phenomena at the Large Hadron Collider (LHC) requires a lot of data O(100 TB/s).
- Sizable filtering (fast and accurate) is necessary, i.e. **the trigger system**.

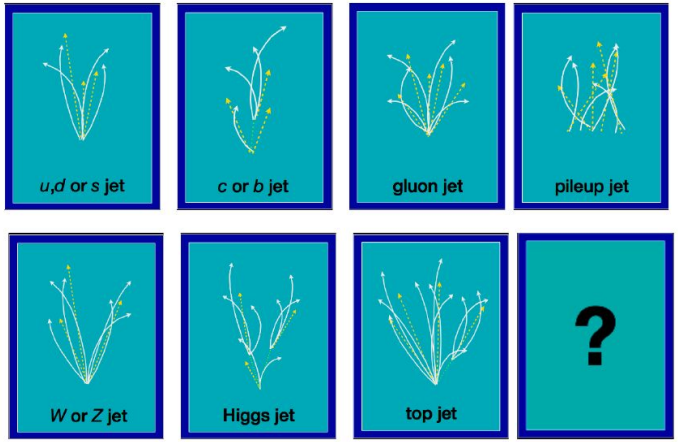
  At the Compact Muon Solenoid (CMS) experiment:

  - L1: from 40 MHz to 100 kHz, processing time **O(1 μs)**, hosted on FPGAs, ASICs.;
  - HLT: from 100 kHz to 1 kHz, processing time **O(100 ms)**, hosted on CPUs.
- The LHC will increase number of collisions per event from 40 to 200, i.e. **HL-LHC**.
- Traditional **reconstruction algorithms** may have scaling issues.

- *Idea 1: replace standard computing with neural networks where feasible.*
- *Idea 2: simultaneous execution of several tasks.*
- *Idea 3: improve latency of neural networks.*

Jet Images

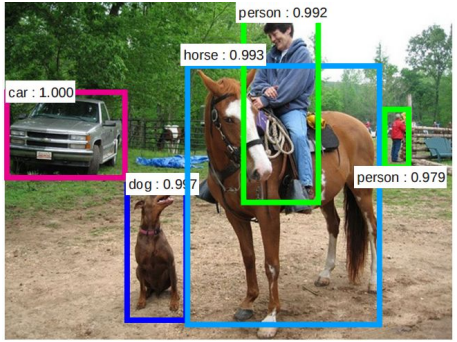
- A ***jet***: collimated shower of particles with adjacent trajectories.
- ***Jet tagging***: classifying the origin of the jet, i.e. the initiating particle.
- Other reconstruction tasks: localization, energy, momentum, mass.
- Traditional tagging: expert features based on energy deposition patterns.
- ***Jet images*** (e.g. 1407.5675): projecting lower level detector measurements into an **image**.
- Enabling use of computer vision techniques, e.g. CNNs.



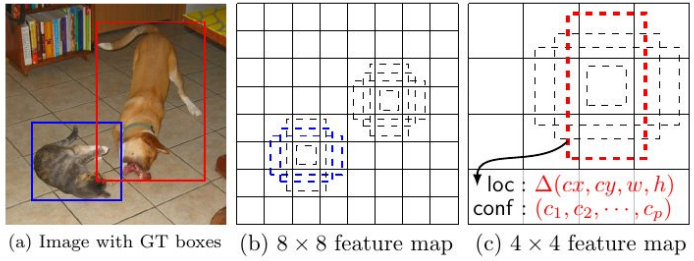
Credit: Nhan Tran

Single Shot Detection

- **Object detection**: a computer vision problem; classification and localization.
- Applications: image annotation, face recognition, pedestrian detection etc.
- One stage detectors: simultaneous execution of tasks.
- Single Shot Multibox Detector (SSD) [1512.02325], a one-stage detector:
  - simple, base network and extra layers use convolutional blocks,
  - localization, classification (and auxiliary tasks) in one forward pass,
  - default regions, aspect ratios and scales are pre-defined.
  - in training refine each box with offsets (cx, cy, w, h),
  - remove duplicates at inference.

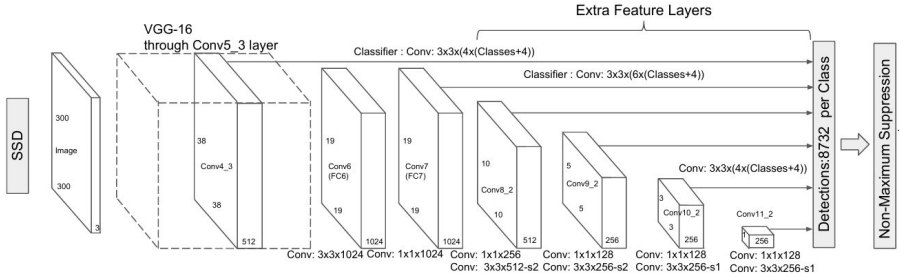


Credit: 1506.01497



(a) Image with GT boxes (b) 8×8 feature map (c) 4×4 feature map

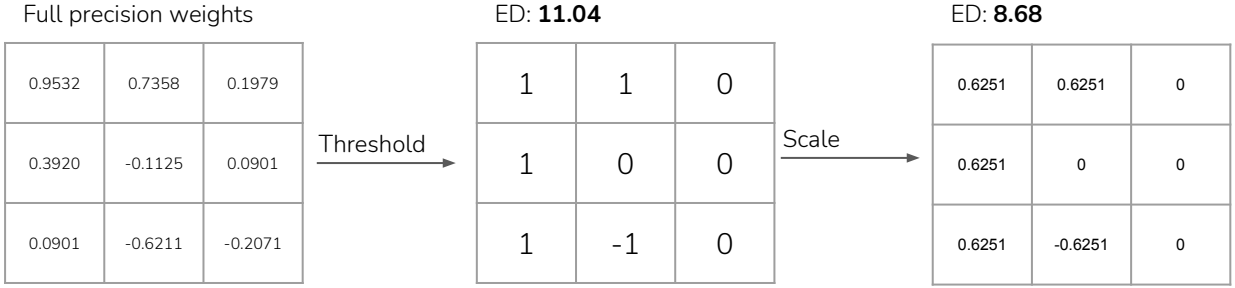
Credit: 1512.02325



Credit: 1512.02325

Ternary Weight Network

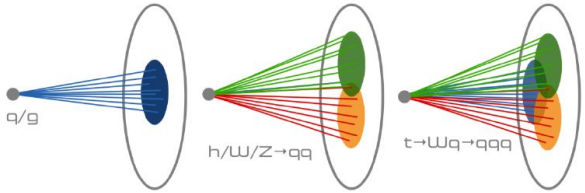
- Unoptimized models often need considerable storage and computational power.
- **Efficient inference**: pruning, compact design, knowledge distillation, low-rank factorisation, **quantization**.
- **Quantization**: reduce precision of operations.
- **Ternary Weight Network** (TWN) [1605.04711]:
  - reduce 32-floating point weights to {-1, 0, 1} during training,
  - scaling factor minimizes the Euclidean distance between full precision and quantized weights.

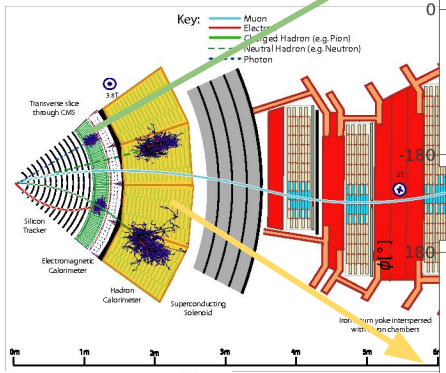


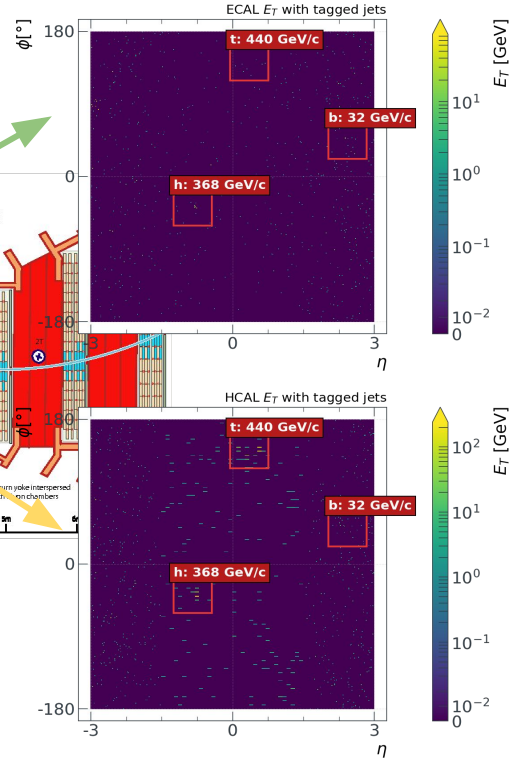
Dataset

- **Input**: calorimeter energy measurements.
- Inner surface of calorimeters is unfolded.
- Crystals represented as image pixels (340×360).
- Two CMS calorimeters used: ECAL, HCAL (Pythia/Delphes): 2 channels.
- Input limited to barrel and endcap, i.e. $|\eta| \leq 3.0$.
- No preprocessing, just maximum scaling.
- Train/validation/test: 90k/30k/90k.
- **Output**: 3 target classes (*bb̄*, *HH/WW*, *or tt¯*), localization and mass.

**bb̄* is a proxy for QCD jets



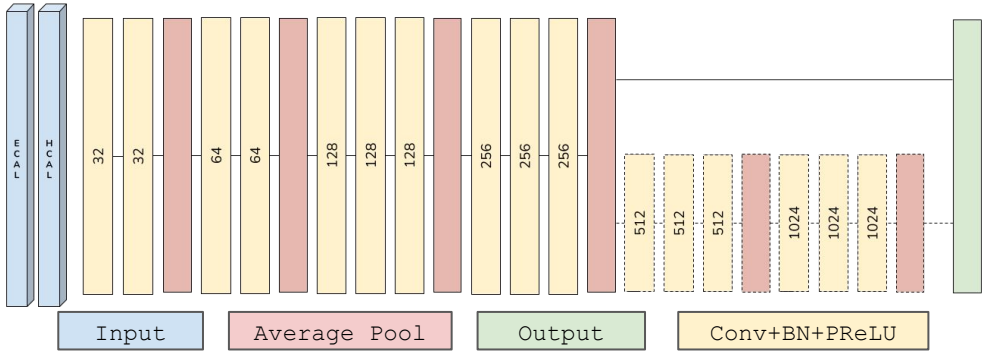




Credit: 1908.05318

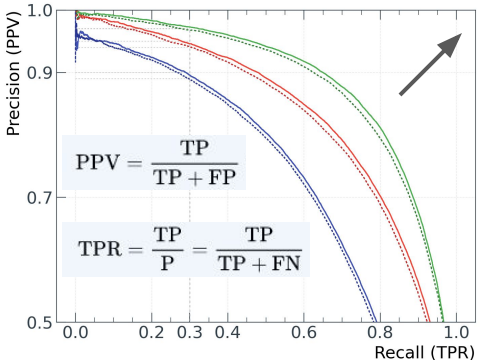
Model, Implementation, Training Procedure

- SSD model modifications:
  - in base network exclusively 3x3 kernels no bias (limitation of chosen hardware),
  - batch normalization added, PReLU activations, smaller channels;
  - removed extra layers;
  - one aspect ratio, one size, two regression outputs (cx, cy offsets) for anchor boxes;
  - 8G OPS.
- Trained with PyTorch, mixed-precision: **github.com/AdrianAlan/jet-ssd**
- TWN Training: warmup with full precision weights.

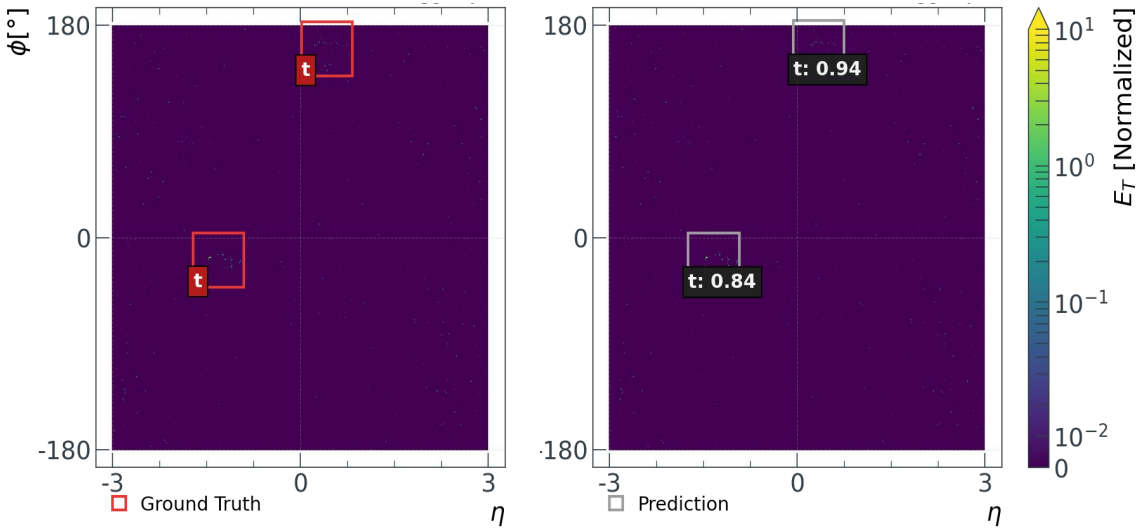


Results: Classification

- Remarkable agreement between TWN and full precision network.
- 1% difference in average precision (AP).
- Localization works well even close to image edges in ϕ.

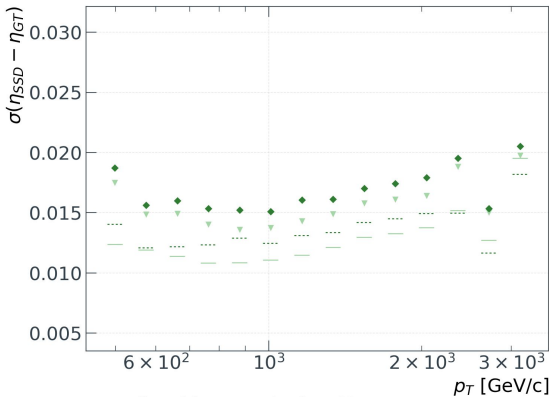


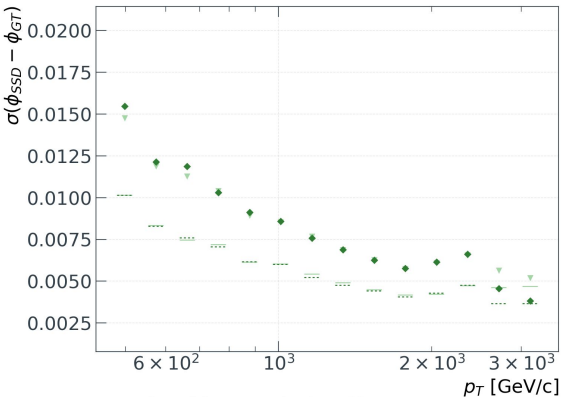
- Full Precision Network: b jets, AP: 0.710
- Full Precision Network: W-H jets, AP: 0.828
- Full Precision Network: t jets, AP: 0.885
- Ternary Weight Network: b jets, AP: 0.702
- Ternary Weight Network: W-H jets, AP: 0.819
- Ternary Weight Network: t jets, AP: 0.876

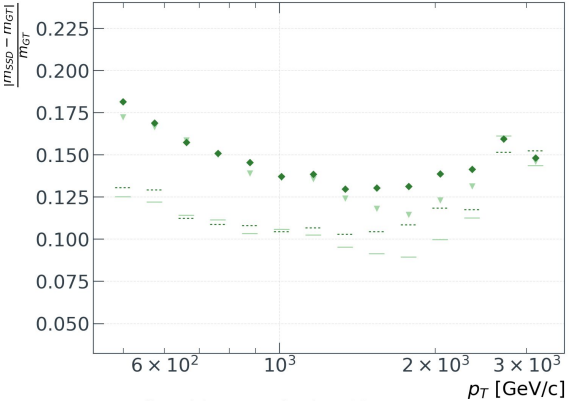


Results: Localization and Mass Regression

- TWN and full precision network localization and mass regression very close.







Full Precision Network: t jets, Q2
Full Precision Network: t jets, μ
Ternary Weight Network: t jets, Q2
Ternary Weight Network: t jets, μ

Conclusions and Future Work

- Jet-SSD can simultaneously tag, locate and regress mass of jets.
- Compressed model with ternary weights retains most of accuracy of full precision equivalent.

WIP:

- Testing latency of Jet-SSD on dedicated ASIC and GPU/TensorRT.
- 3-channel version of Jet-SSD (with tracker).
- Jet-SSD can be further improved with heterogeneous quantization or hardware aware training.

Acknowledgments
A. A. P., M. P., S. S. and V. L. are supported by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation program (grant agreement 772369). A. A. P. is supported by CEVA under the CERN Knowledge Transfer Group. V. L. is supported by Zenseact under the CERN Knowledge Transfer Group. We thank Simons Foundation, Flatiron Institute and Ian Fisk for granting access to computing resources used for this project.