Convolutional Neural Networks for photon ID

S. Noacco Rosende, F. Monticelli

[1] Instituto de Física La Plata (CONICET-UNLP)

[2] Universidad Nacional de La Plata

[3] ATLAS CERN

June 30, 2021

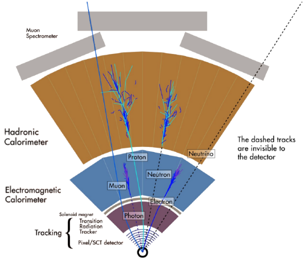
IFLP CONICET UNLP — UNIVERSIDAD NACIONAL DE LA PLATA

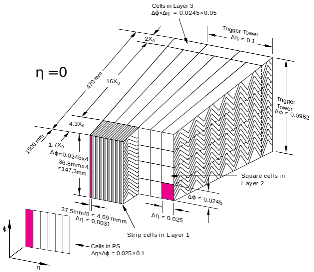
Context

- The correct identification of photons in ATLAS is crucial for many very important physics analysis and new physics searches.
- Photon reconstruction and identification is performed using cut-based selection based on information from the Electromagnetic Calorimeter (EMCalo).
- Why **D**ense **N**eural **N**etworks in general?
  - Potencial performance improvement in very specific tasks, such as classification.
  - Allows for the direct use of low level data, therefore possibly reducing sistematics errors.
  - Potencial speed at predictions improvements, specially when optimized for hardware.
- Why **C**onvolutional **N**eural **N**etworks?
  - Proven across domains its power to classify events from an image input.
  - We'll see that the cells play a fundamental role in the reconstruction and identification of objects, but there is a heavy processing involved for their implementation, CNNs reduce that processing to a minimum.

Photons' interaction with the detector

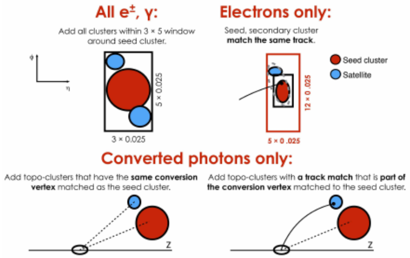
- Photons interact with the detector leaving most of its energy in the 2nd Layer of the EMCalo. Since Photons and Electrons leave a similar trace in the detector, we mostly care for the Tracking and EMCalo portions of the detector.
- The **Tracking** is key for determining convertion verteces, which are used in the Offline Reco&Id to discriminate between Converted and Unconverted photons.
- The **Electromagnetic Calorimeter** (EMCalo):
  - Has a cylindrical geometry divided in two regions: barrel ($|\eta| < 1.37$) and end-cup ($|\eta| > 1.52$), and in-between we find a transition region that is not used for photons.
  - It is segmented in three layers of cells:
    - Lr1: finely segmented in η direction.
    - Lr2: collects most of the energy deposited by photon/electron showers ($17\ X_0$)
    - Lr3: correct for leakage beyond the EMCalo for high-energy showers.

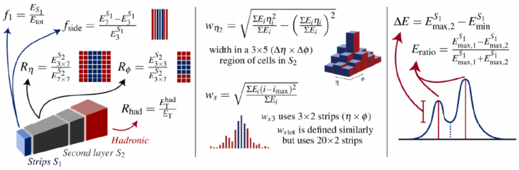




Photon reconstruction and identification

- The reconstruction is implemented under a *SuperClusters* algorithm for Run3 and Offline Run2, while for Run2 at **H**ight **L**evel **T**rigger, it is done using *Sliding Window* algorithm. However in both cases the raw data used is the same, the ROI cells, and track info when available.
- The current technique for identification is based on applying cuts to variables describing the shower shape of the interaction between the candidates and calorimeter, and isolation criteria. A set of cuts on this variables is known as a WorkingPoint (WP).

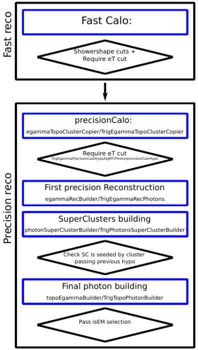




Photon reconstruction and identification

The photon Id is done both at OFFLINE as well as at Trigger Level HLT (ONLINE), with significant differences.

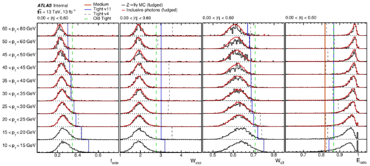
- The photon Reco and Id at HLT is carried out in two steps:
  - FastReco: optimized for early bkg rejection. Currently uses SS cuts and E_T cut
  - Precision: instantiating offline algorithms. assumes all γ are Unconverted, The cells data is not *a priori* the same as Offline.



- While at OFFLINE (described by Fran in the past presentation): corrects for pileup, and it's optimized for convertion.

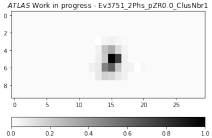
Category	Description	Variable	loose	tight
Acceptance	$\lvert\eta\rvert < 1.37 \cup 1.52 < \lvert\eta\rvert < 2.37$		+	+
Hadronic leakage	Ratio of E_T in the first layer of the hadronic calorimeter to E_T of the EM cluster (used over the range $\lvert\eta\rvert < 0.8$ and $\lvert\eta\rvert > 1.37$)	R_{had1}	+	+
	Ratio of E_T in the hadronic calorimeter to E_T of the EM cluster (used over the range $0.8 < \lvert\eta\rvert < 1.37$)	R_{had}	+	+
EM Middle layer	Ratio of 3×7 $\eta \times \phi$ to 7×7 cell energies	R_η	+	+
	Lateral width of the shower	$w_{\eta 2}$	+	+
	Ratio of 3×3 $\eta \times \phi$ to 7×7 cell energies	R_ϕ		+
EM Strip layer	Lateral shower width calculated from three strips around the strip with highest energy deposit	w_{s3}		+
	Total lateral shower width	w_{tots1}		
	Energy outside the core of 3 central strips but within 7 strips divided by energy within 3 central strips	F_{side}		+
	Difference between the energy associated with the second maximum in the strip layer and the energy energy reconstructed in the strip with minimum value found between the first and second maxima	ΔE		+
	Ratio of energy difference associated with the largest and second largest energy deposits to the sum of these energies	E_{ratio}		+

Table 2: Discriminative shower shape variables used for *loose* and *tight* photon identification.



Cluster data

- A Cluster is simply a collection of cells matched to an object candidate. And in all cases the center cell contains the highest amount of energy of the cluster.
- Differences in Cluster data:
  - between object candidates: *egamma* uses a different *Sliding Window* algo than *jets* and E_T^{miss}, the latter uses the full calorimeter basically.
  - between algorithms: *Sliding Window* generates fixed size clusters. *Topocluster* (or the *SuperCluster* variant) generates un-fixed size clusters.
  - between detector regions: sizes of cells and Clusters vary within a subdetector (e.x. from one Layer to another), and between the barrel and endcap region.
- Unsolved-known-issue: some Cluster have *gaps* where cells are missing, or *misalignments* where a set of cells are shifted.
- In all the following developments, all cells from a given cluster are **normalized** to the sum of all cluster cells energy, i.e. the total cluster energy. This serves the purpose of removing any E_T bias at the training stage.
- All this consideration come to show that there is plenty of room for improvement and optimization when dealing with cells.

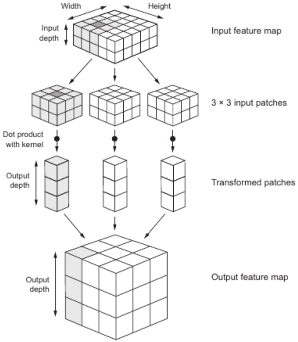


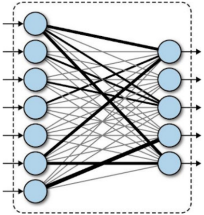
CNN overview

- A DNN model $\mathcal{M}$ is a non-linear function, $\mathcal{M}(\vec{x}) = A \cup W$
  - Architecture A: is a graphs where its nodes preform defined operations.
  - Matrix of weights W: determines the relevance of each operation.

  It all comes together through the Backpropagation algorithm.
- At its core a CNN is built from at least two Neural Networks Blocks (or sub-graphs):
  - The Convolutional block is made of layers where the nodes compute a **convolution operation** (conv-op) between a filter and the layer-input.
    - filters: initialized randomly and modified through training.
    - feature maps: conv-op output, compresses the relevant info extracted from the data.
    - activation function: enhance filter relevance and adds a non-linear resource.

    It **extracts the most relevant features** from the input, so essentially it plays the role of the SS; this doesn't mean that a CNN approach is incompatible with a SS approach.
  - The Dense block plays as always the role of classification.
- The connected architecture together with the backpropagation training, fully correlates the two blocks to give a set of weight that optimized the election of Convolutional Filters, and Decision Weights to provide a high performance non-linear decision boundary.





IFLP CONICET UNLP · UNIVERSIDAD NACIONAL DE LA PLATA

The CNN dev-road

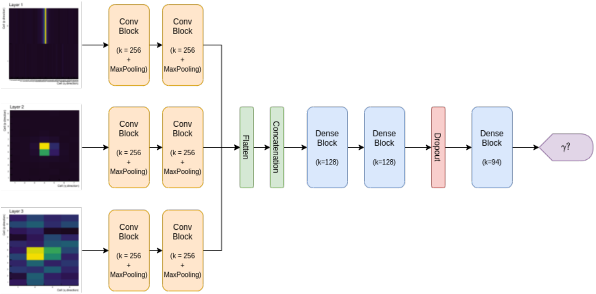
1. ***Offline CNN development:*** Mohamed Belfkir developed a CNN framework to identify photons at the OFFLINE stage, using the CaloClusters MC data from the 3 layers of the EMCalo. In his study he shows the perfomance of the model with Inclusive, Converted and Unconverted Photons.
2. ***HLT CNN development with Data:*** In parallel I developed (in the context of my Master Thesis), a CNN model to identify photons at the FastReco step of the HLT, based on Data. One of the mayor difficulties we ran into was the scalability of the infrastructure.
3. ***Development convergence, improvements and extension:*** As part of my QT, I extended and improved Mohamed's framework to train a model at the FastReco step of the HLT using MC data. This framework now overcomes the previous limitation of infrastructure scalability since it allows to train and validate using batch-system resources. The next objective we have is to implement and compare the Offline model in the Precision Step. This opens a world of posibilities, specially in the compatibility and consistency between Offline and Online.

CNN model for Offline Photons Id

Addressing the 1 item of the dev-road. Data sets used: **MC data** from ZllyAthDerivation, by Pythia8.

- ***For background only:*** **DiJet** (14 Gb), MC16 list contains: `JF17, JF23, JF35, JF50`; where `JFx` denotes a QCD dijet sample with a generation level cut at $x\ GeV$.
  Signal events will be *TruthMatch* photons.
- ***For signal only***: **GammaJet** (33 Gb), MC16 list contains: `DP8_17, DP17_35, DP35_50, DP50_70, DP70_140, DP140_280`; where `DPx_y` denotes a Direct Photons sample with a generation level cut between $x\ GeV$ and $y\ GeV$.
  Background events will be *Non TruthMatch*

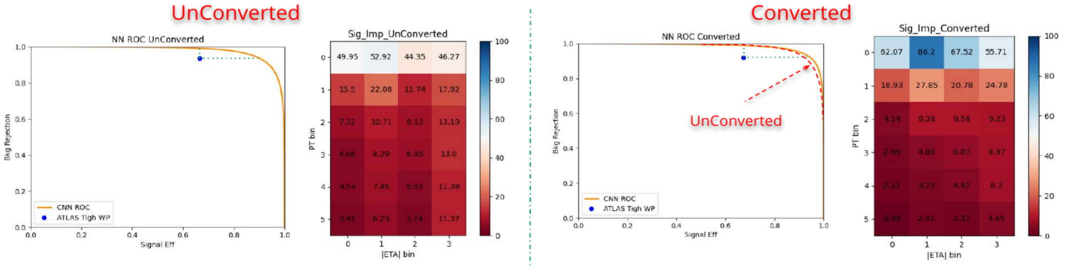
- Models' architecture:



- Training and validation conditions:
  - $p_T > 10\ GeV$, $|\eta| < 2.5$, Isolation = Tight, Target = "TruthMatch".
  - Inclusive training in η and p_T, optimized for Converted and Unconverted.
  - Data set split: 50% training, 25% validation and testing.
  - Trained for 20 epochs, drop rate 8%, learning rate $1e-4$.

CNN model for Offline Photons Id

Performance of the model on testing MC dataset. We show the inclusive (in η and p_T bins) ROC curves, next to the binned AUC improvement agains Tight WP.

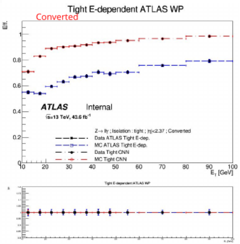


- On the left for Unconverted photons and on the right for Converted photons.
- In both scenarios the model outperforms the current WP, where in the converted case the differences are greater.
- It can be seen that performance diminishes towards high η and p_T.

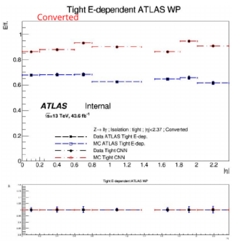
CNN model for Offline Photons Id

Results computed on 2017 data [1] at the same background rejection level as ATLAS Offline Tight WP:

- Converted:

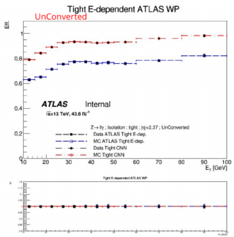


(a) E_T

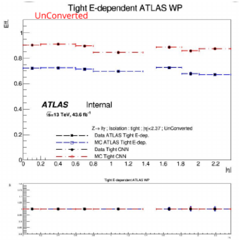


(b) $|\eta|$

- Unconverted:



(a) E_T



(b) $|\eta|$

- CNN model out-preforms Offline Tight WP at same Bkg rejection level by arround 20% margin in both η and p_T, for Converted and Unconverted photons.
- This model is already deployable in ATHENA through the ONNX framework.

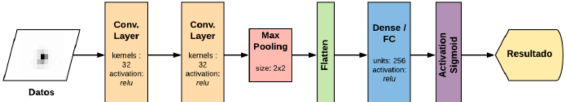
[1] references Update on CNN PhotonID with RadiativeZ (indico 965633) and Photon Identification Efficiency With Convolutional Neural Network (indico 1009062)

CNN model for FastReco Photons Id

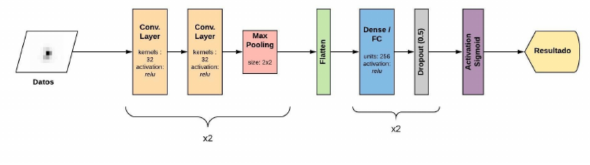
Addressing the 2 item of the dev-road. Data sets used ($\simeq 60k$ pure events):

- **For signal only** EGAM3 file 2018 at $\sqrt{s} = 13\ TeV$.
  `data18_13TeV.00356205.physics_Main.deriv.DAOD_EGAM3.f956_m2004_p3583`.
  Signal events are photons from Zrad.
- **For background only**: xAOD file 2018 at $\sqrt{s} = 13\ TeV$.
  `data18_13TeV.00358615.physics_Main.merge.AOD.f961_m2020/data18_13TeV.00358615.physics_Main.merge.AOD.f961_m2020._lb0666`.
  Background events come from an inversion of the isolation criteria.

- Models' architecture:

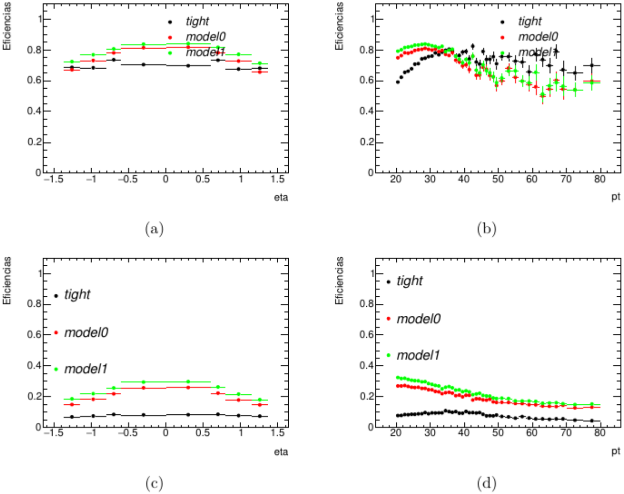


(a) model 0



- Training and validation conditions:
  - Both carried out on Data Samples, not MC.
  - Inclusive training in η and p_T.
  - Data set split: 55% for training,25% validation, 20% testing. 40 epochs training.
  - optmizer = RMSprop, learning-rate = 1e-4,
  - We defined 2 models, to adress the impact of model complexity.

CNN model for FastReco Photons Id



(a) (b)

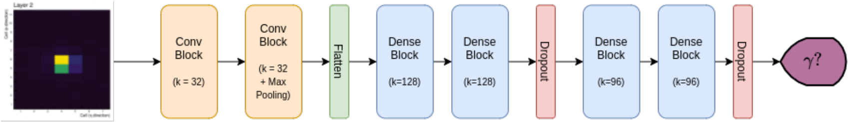
(c) (d)

- Offline Tight WP Bkg rejection is better than the CNN model for a 15% margin.
- CNN model preforms over 10% better in the η scope agains Offline Tight WP, while remaining comparable in the p_T scope.
- TODO train and validate on larger DS, with the new framework.
- TODO compute Signal Efficiencies for a Bkg rejection fixed to the Tight WP Bkg rejection level.

CNN model for FastReco Photons Id

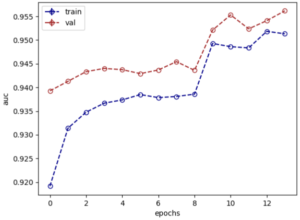
Addressing the 3 item of the dev-road. Data sets used: same as for CNN Offline Photons Id.

- Models' architecture:

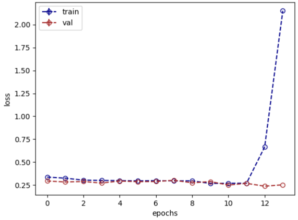


Training and validation conditions:

- - Only using MC data.
  - Training inclusive in η and p_T, validation is binned.
  - Architecture configured dynamicly (Keras Tuner).
  - Trained for 40 epochs (with Stop-Loss enabled), in batches.
  - Optimizer = Adam, learning-rate = 1e-4



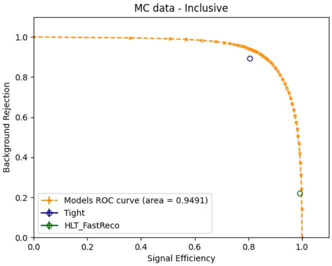
(a)



(b)

CNN model for FastReco Photons Id

Results computed on testing MC dataset:



(a)

(b)

- The CNN model out-preforms current HLT Precision step Tight WP, for every (η, p_T) bins, in the context of an inclusive training (not optimized for neither η nor p_T)
- The CNN model is comparable to the preformance of the Offline Tight WP in many (η, p_T) bins, and it out-preforms it in the Inclusive scope.
- TODO: compute Efficiencies at fixed Bkg rejection on data.

IFLP CONICET UNLP — UNIVERSIDAD NACIONAL DE LA PLATA

On-going and next steps

Some of the pending tasks where already mentioned in the previous slides, here we extend that list:

- For the model at HLT Precision Step
  - Implement the Offline-CNN model at the HLT Precision step. We would like to test:
    - trained on Offline-photons and validated on HLT-photons, does it extrapolate well?.
    - trained on HLT-photons and validated on HLT-photons, consistency between Trigger and Offline.
  - Compute Efficiencies at fixed Bkg rejection on data.
  - Deploy the HLT model to ATHENA.
- For the model at HLT FastReco Step
  - Train & validate on larger MC dataset, using the new framework.
  - Use FastReco cells as input.
  - Right now is on stand-by, because there is a current DNN implementation well developed, the *Ringer Neural Network*.
- For all models:
  - Train models on Data, when possible, for low p_T ZRad can be used.
  - Explore options to optimize training for η and p_T.
- Caveats about NNs in general:
  - All biases and systematics need to be controlled and understood, in order to make use of any identification.
  - Training even if done in MC, are cross-checked in real photons in real DATA using ZRad.
  - Training with real photons is unavailable for high p_T.
  - Background estimation is tricky:
    - With Cut-based approach one can obtain a data-driven backgound enriched sample by revering some cuts.
    - With NN selection work is in progress.

UNIVERSIDAD NACIONAL DE LA PLATA

Summary

- We've seen and understood the potential of Convolutional Neural Network in the task of classification from low level data (Clusters).
- Although most results are contemplated in a preliminary scope, big improvements in efficiencies and overall performance have been achieved.
- There is a strong effort being input in the development, optimization, deployment and validation of DNN techniques in many fronts of the ATLAS teams.
- Many issues to address in the near future to have this methods implemented:
  - Training and Validating on Data.
  - Computing systematics.
  - Deployment is still in its early development stages.
  - Further improvement in data infrastructure to handle Trainings.
  - Accessing the clusters raw data (cells) is not trivial, and not always available.