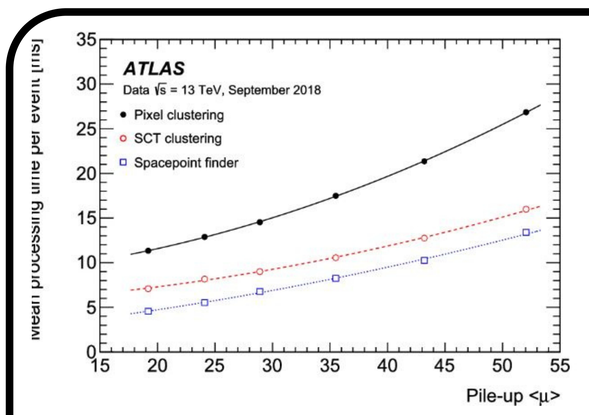
Un nuovo approccio per assistere la ricostruzione veloce delle tracce a livello del trigger dell'esperimento ATLAS

Alessandro Zaio, Andrea Coccaro, Carlo Schiavi
Incontri di Fisica delle Alte Energie 2024, Firenze
Università di Genova

A livello di High Level Trigger (HLT) la ricostruzione delle tracce, o tracking, è il processo che richiede la maggior quantità di risorse CPU.
Gli algoritmi di tracking comprendono due step sequenziali:

- il Fast Tracking seleziona tra i punti spaziali identificati da pixel e strip, terne compatibili al passaggio di un traccia, che viene poi estesa ai layer restanti.
- il Precision Tracking, seleziona le tracce di migliore qualità e vi applica un fit.

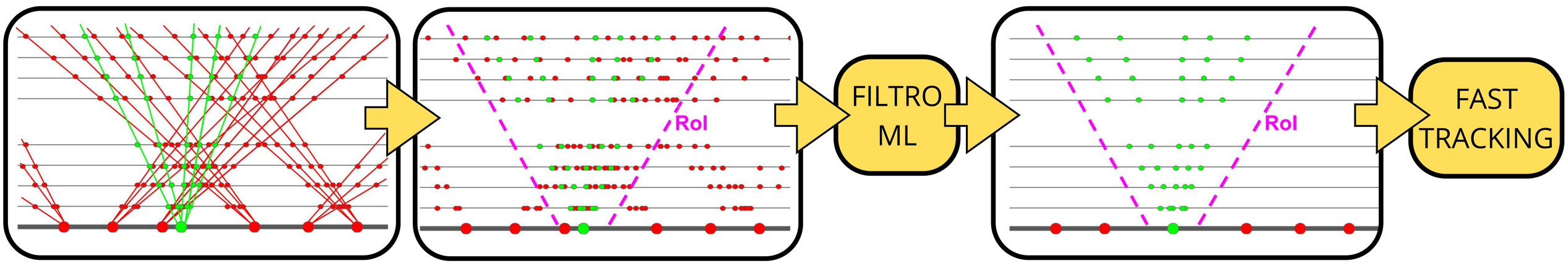
Durante Run 2 è stata osservato che la tempistica del Fast Tracking dipende secondo una legge di potenza dal pile-up, definito come il numero di interazioni *pp* associate a un'unica collisione tra bunches.



Fonte: ***"The ATLAS Inner Detector Trigger performance in pp collisions at 13 TeV during LHC Run 2"***, Eur. Phys. J. C 82 (2022) 206.

Con l'upgrade a High Luminosity-LHC (HL-LHC) sono attesi valori del pile-up medio $\langle\mu\rangle$ nell'intervallo tra 140 e 200 collisioni *pp,* per i quali l'attuale farm di CPU che gestisce HLT non riuscirebbe a sopportare il costo di computing.
È quindi in corso un ripensamento delle strategie di trigger, che potrebbe anche vedere l'utilizzo di hardware eterogeneo, includendo GPU e/o FPGA.

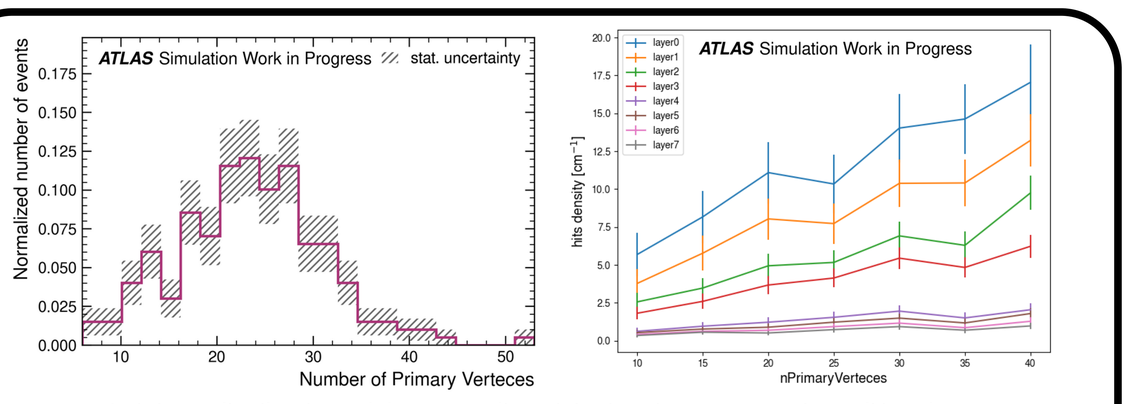
Proponiamo un algoritmo di Machine Learning (ML) che riceva i punti spaziali ottenuti dall'Inner Detector (ID) e impari a filtrare dalla regione d'interesse (o RoI) i punti dovuti alle tracce di pile-up, in modo da ridurre il combinatorio all'interno del Fast Tracking.



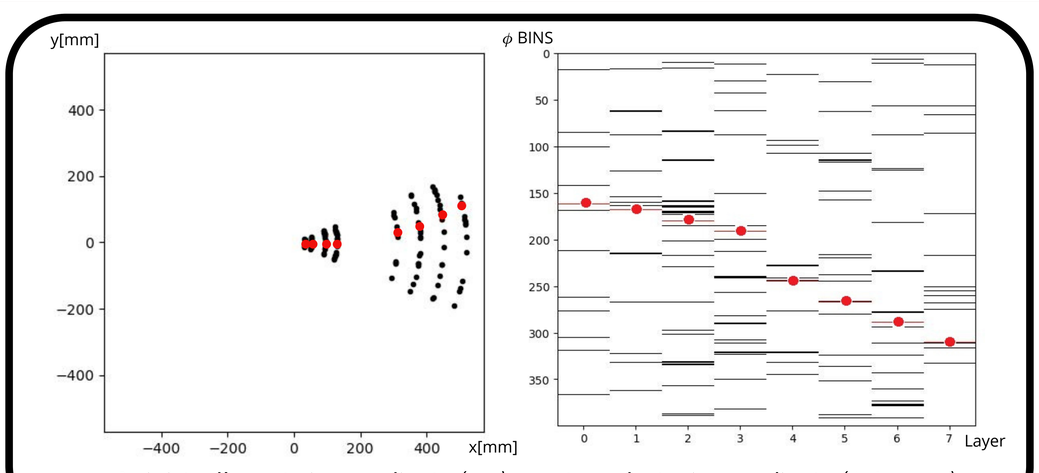
Per testare la validità di questa idea iniziamo allenando e testando l'algoritmo su degli eventi sintetici generati attraverso un modello semplificato.
Gli hit che includiamo negli eventi sono divisi in due categorie:

- SEGNALE, ottenuti intersecando tracce cariche con i layer dell'ID
- FONDO: Hit di rumore generati senza correlazione tra layer diversi.

Per avvicinarci ad una descrizione più realistica consideriamo un sample Monte Carlo di jet $t\bar{t}$ e calcoliamo la densità di hit per layer e al variare del pile-up.
Usiamo questi parametri per ottenere eventi corrispondenti a diversi valori di $\langle\mu\rangle$.



A sinistra: distribuzione del numero di vertici primari per un campione di jet ttbar.
A destra: densità di hit per layer in funzione del pile-up medio degli eventi

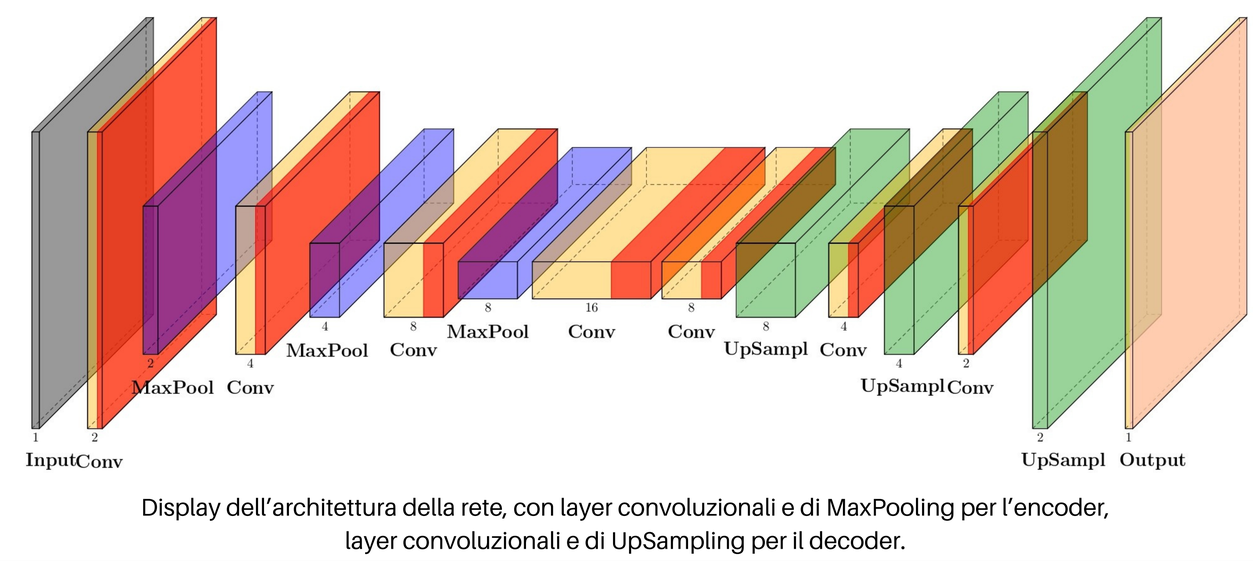


A sinistra l'evento in coordinate *(x, y),* mentre a destra in coordinate *(ϕ, Layers,)*.

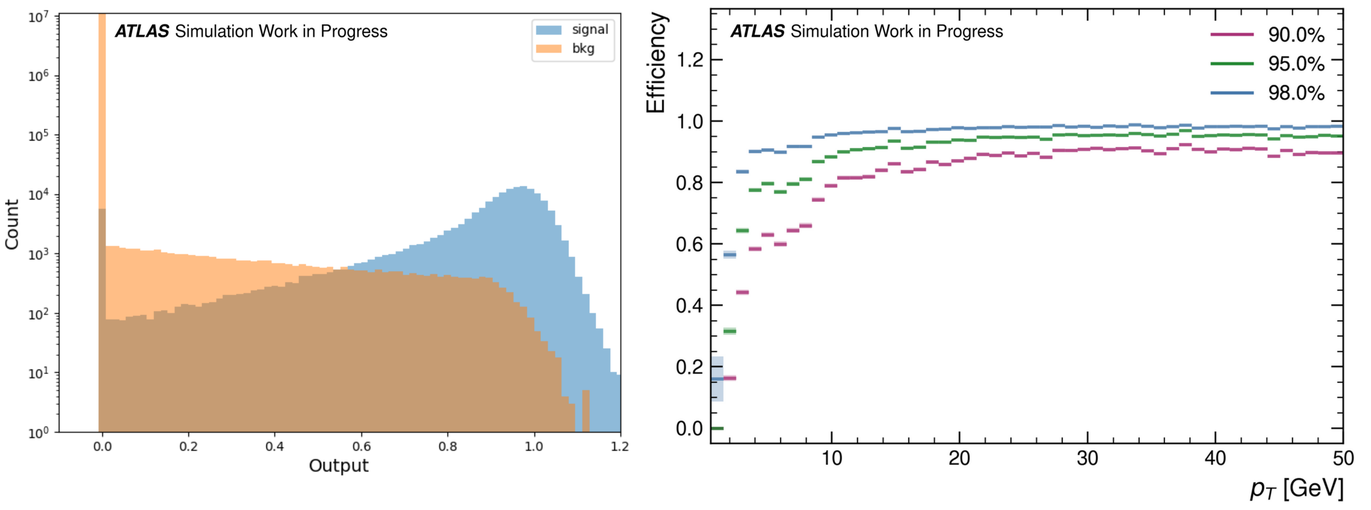
L'algoritmo che utilizziamo è una CNN o Rete Neurale Convoluzionale, poiché questi modelli sono facilmente accelerabili su schede FPGA tramite l'utilizzo di software commerciali.
Per convertire gli eventi in immagini, proiettiamo gli hit generati in un cono di ΔR=0.4, sul piano *xy*. Cambiamo coordinate passando da *(x, y)* a *(ϕ, Layers,)* scegliendo come larghezza dei bin in ϕ il valore 0.02 *rad*, che corrisponde a 68 μm nel primo layer e circa 1 *mm* nell'ottavo. Le matrici in input alla rete avranno entrate con valore 1 se almeno un hit è presente in quel bin, altrimenti valore 0.

Vogliamo che l'algoritmo fornisca in output un immagine con entrate a valore 1 in corrispondenza degli hit di segnale e invece 0 per gli hit di fondo. Prendiamo ispirazione dall'architettura dei Denoising Autoencoders, caratterizzata da una compressione dell'input nella fase dell'Encoder e il ritorno alla taglia dell'input nella fase del Decoder
Utilizziamo un modello molto leggero, con soli 12 mila parametri, particolarmente adatto per un'inferenza veloce.

$$L = \frac{1}{N}\sum_i (y_i - \bar{y}_i)^2$$ MSE loss calcolata sui bin nei quali era presente un hit in input.



Display dell'architettura della rete, con layer convoluzionali e di MaxPooling per l'encoder, layer convoluzionali e di UpSampling per il decoder.



A sinistra: istogramma dei valori dell'output dei bin. La rete acquisisce potere discriminante.
A destra: efficienza nella ricostruzione degli hit di segnale in funzione di p_T.

La rete è allenata su 1 milione di eventi a singola traccia di segnale con p_T compreso tra 0.5 e 50 *GeV* utilizziamo come discriminante l'output di ciascun bin e osserviamo che la rete ha appreso il task.

Cosideriamo l'efficienza nel ricostruire hit di segnale e fissiamo il Working Point testando su un sample di eventi a singola traccia ad con p_T tra 40 e 50 *GeV*.
La performance della rete peggiora sensibilmente per tracce a basso, questa caratteristica può essere utile per filtrare le tracce di pile-up a basso momento.

Alessandro Zaio
alessandro.zaio@ge.infn.it
Università degli Studi di Genova
INFN Sezione di Genova