Using Machine Learning techniques to determine the impact parameter of collisions in the CBM experiment

Manjunath Omana Kuttan, Jan Steinheimer, Kai Zhou, Andreas Redelbach, and Horst Stoecker

SPONSORED BY THE
Federal Ministry
of Education
and Research

FIAS Frankfurt Institute for Advanced Studies

ErUM-Data IDT

GSI

HGS-HIRe for FAIR
Helmholtz Graduate School for Hadron and Ion Research

The DL/ML group at FIAS

Physics: Horst Stöcker,

Kai Zhou (Funded in part by ErUM-Data and by Samson AG donation),

Jan Steinheimer (Funded in part by ErUM-Data and by Samson AG donation),

Kyrill Taradyi (Funded by Stiftung Polytechnische Gesellschaft),

Olena Linnyk (funded by Samson AG donation),

Manjunath Omana Kuttan (PhD Student, funded GSI F&E and Samson AG donation),

Shriya Soma (PhD Student, funded by funded GSI F&E and Samson AG donation).

Other fields include:

➔ smart power grids, seismology, HR, hydrodynamics etc...

Major collaborations

Applications of AI/DL/ML in fundamental research:

- **ErUM data**
  - DL for CBM experiment
- NA61/SHINE
  - Project “Deep learning for fast TPC data processing”
- Xidian-FIAS International Joint Research Centre
  - Long term center founded in 2019 to research for AI in natural and life sciences
- FIAS-SCNU Guangzhou
  - Planned research center on physics related topics, with focus on High Intensity heavy ion Accelerator Facility (HIAF).
- FIAS Huzhou University
  - Collaboration on physics with focus on HIAF
- More to come...

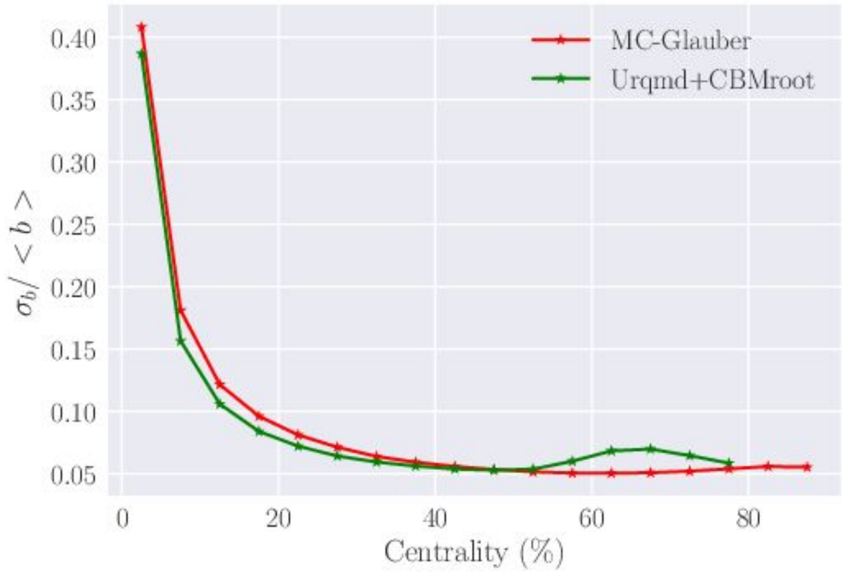
Impact parameter (b) determination in HIC

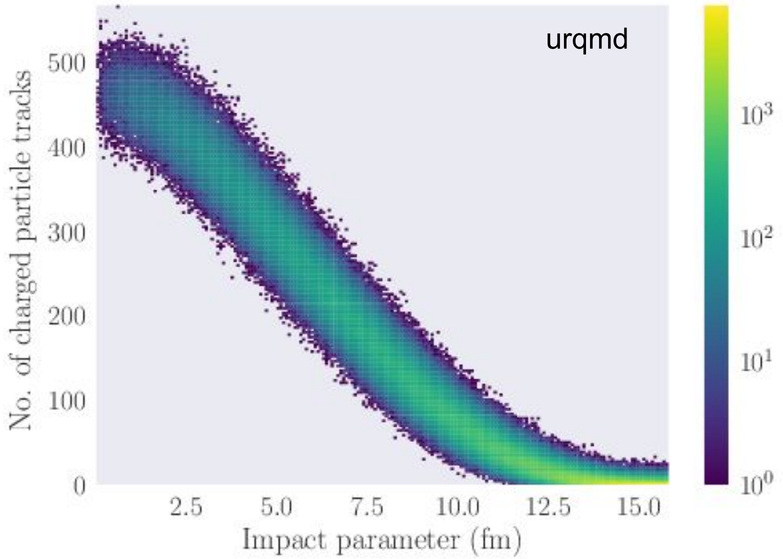
- 'b' is not directly measurable
- Final state observables carry this information
    - eg: charged track multiplicity, spectator energy etc…

Model an observable	Define centrality classes	Find event centrality from the observable
Track multiplicity ↓ Glauber model	Central events → Large multiplicity Peripheral events → Small Multiplicity	Makes a broad classification of events

- We only extract the 'likely' distribution of b (mean and variance) for all events in a centrality bin

‘b’ spread for Glauber model vs. UrQMD
Au+Au @ 10 AGeV





- The likely ‘b’ distribution depends on the model used!
- The accuracy in ‘b’ determination is limited by the spread of the observable used

DL in High Energy Physics

- DL/ ML models are extensively used in experimental HEP
  - Estimation of observables
  - Track reconstruction
  - Particle identification
  - Identification or rare process
  - And more…
- Several studies have also shown b reconstruction using ML
  - Using event generator output
  - Simplified detector definitions
  - Highly processed data

Can DL methods be used to determine the impact parameter of a collision in CBM experiment from raw experimental output?

Gathering data

1 UrQMD 3.4

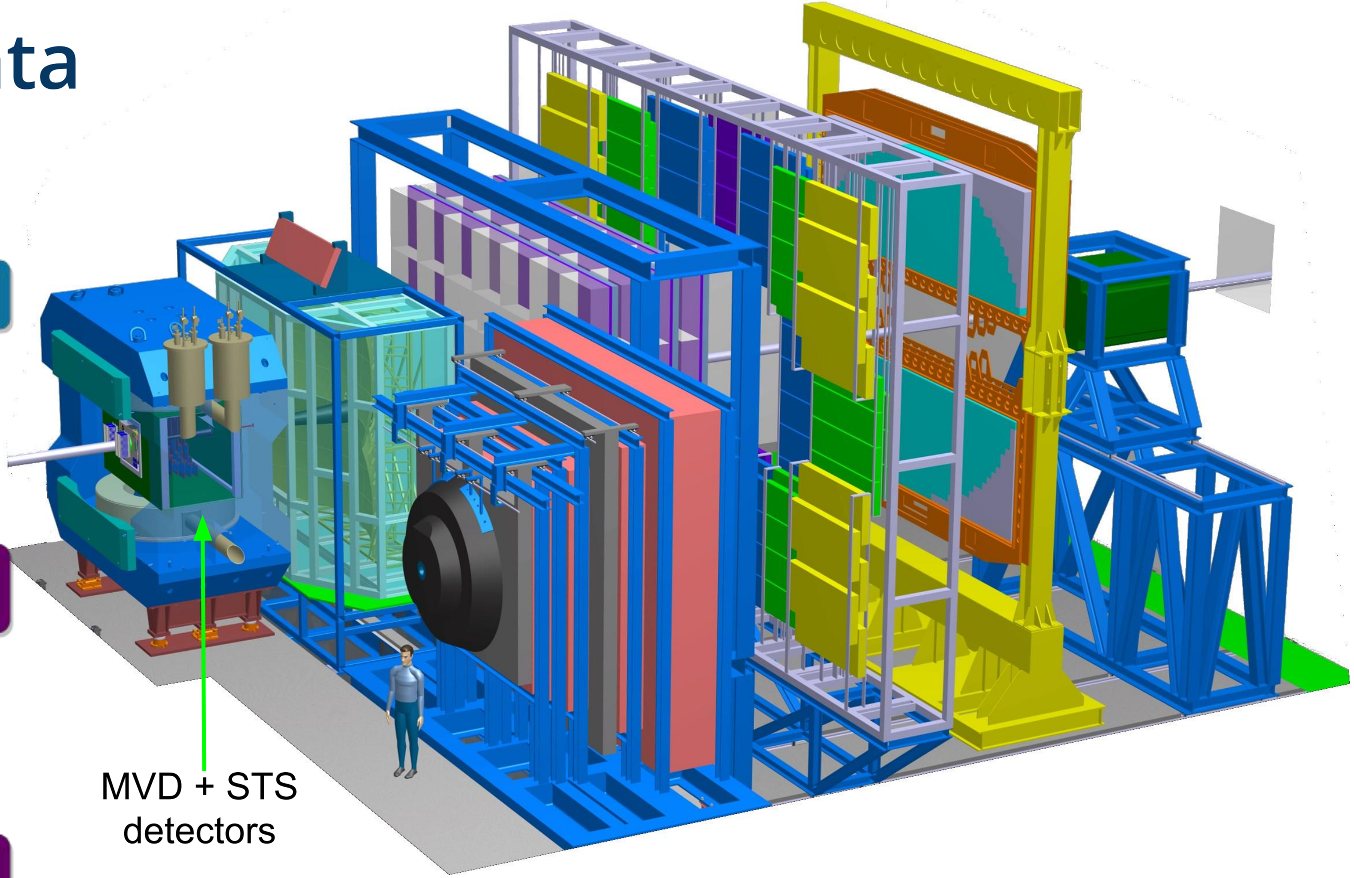
2 Unigen

3 CBMroot : Oct 2019 release

3 Macro: run_transport.C

3 Macro: run_digi.C

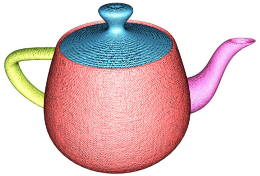
3 Macro: run_reco_event.C

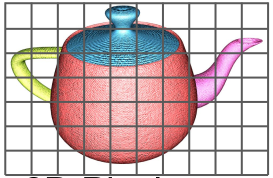


- **Au+Au @ 10 AGeV**

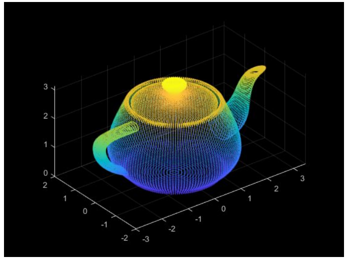
DL model for b reconstruction

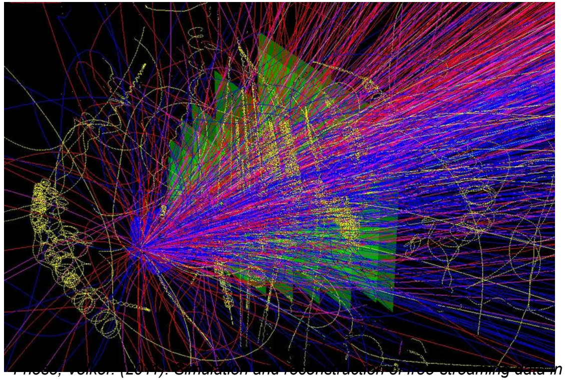
- Experimental data: tracks or hits of particles
    - Each particle is a point in a point cloud
    - The order of input shouldn't affect the output





2D Pixel map





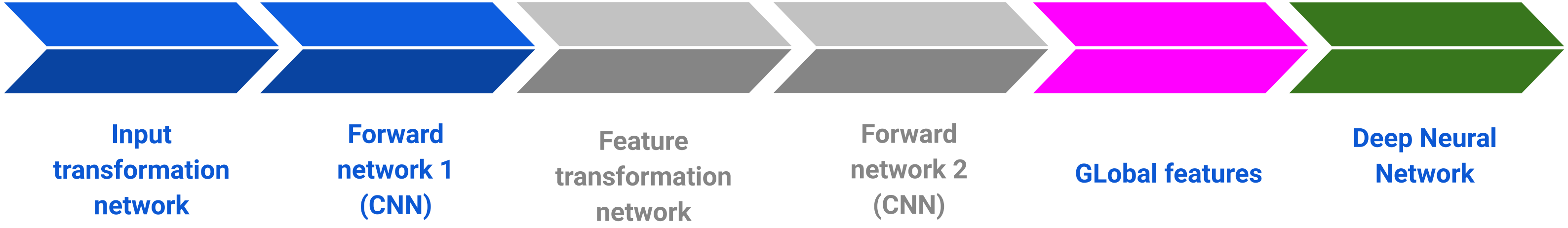
Friese, Volker. (2011). Simulation and reconstruction of free-streaming data in CBM. Journal of Physics: Conference Series. 331. 032008. 10.1088/1742-6596/331/3/032008.

- 3D Voxels map
    - voluminous!
- Pointcloud: Unordered 2D array of (x,y,z) coordinates (or other point attributes) of each point
    - Efficient representation for higher dimensions

X1,y1,z1,.....
X2,y2,z2,.....
X3,y3,z3......
. . .
xn,yn,zn,.....

Pointnet-based DL models

- Pointnet: Deep learning model for point clouds
  - Unordered
  - Invariant to transformations

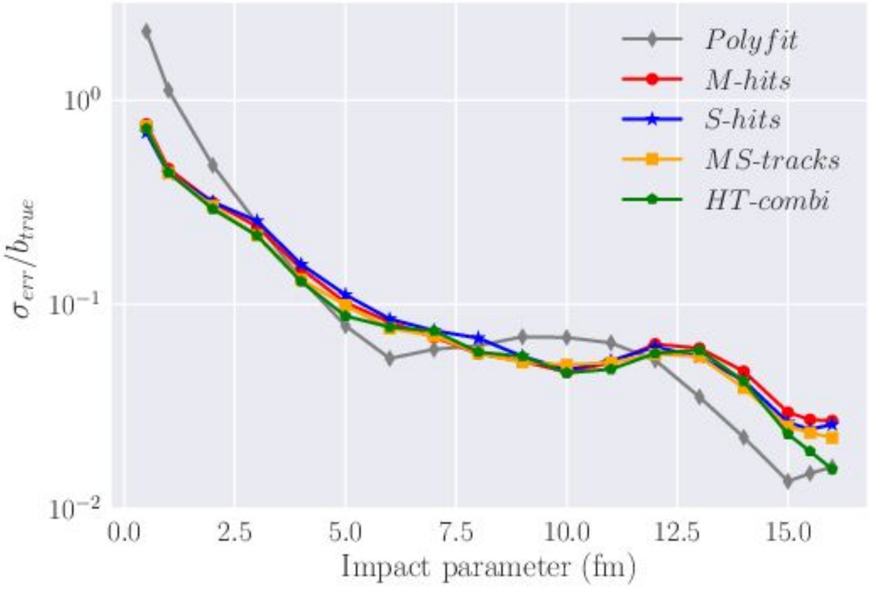


- We have developed pointnet based models which can reconstruct b from 4 different kinds of data

Models

01	**M-hits**	• (x,y,z) of all Hits in MVD
02	**S-hits**	• (x,y,z) of all hits in STS
03	**MS-tracks**	• (x,y,z,dx/dz,dy/dz,q/P) of all tracks in first and last plane from MVD+ STS
04	**HT-combi**	• Hits from MVD and tracks from MVD+STS
05	**Polyfit (non-ML baseline)**	• Third order polynomial fit to multiplicity vs. impact parameter curve

Results: relative precision

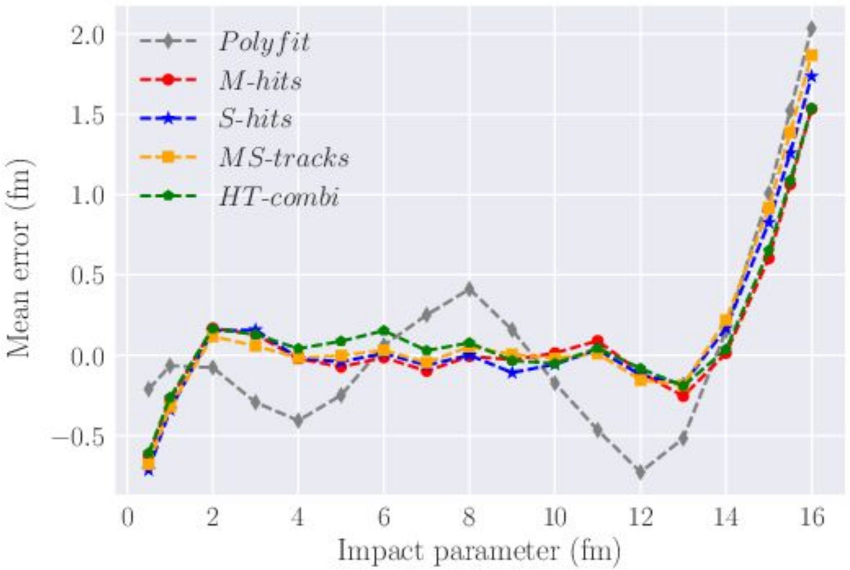


σ_{err} = standard deviation of error in predictions (err= true-predicted)

- Quantifies precision in predictions
- Polyfit fails for central events!
- Similar precision for b>3 fm

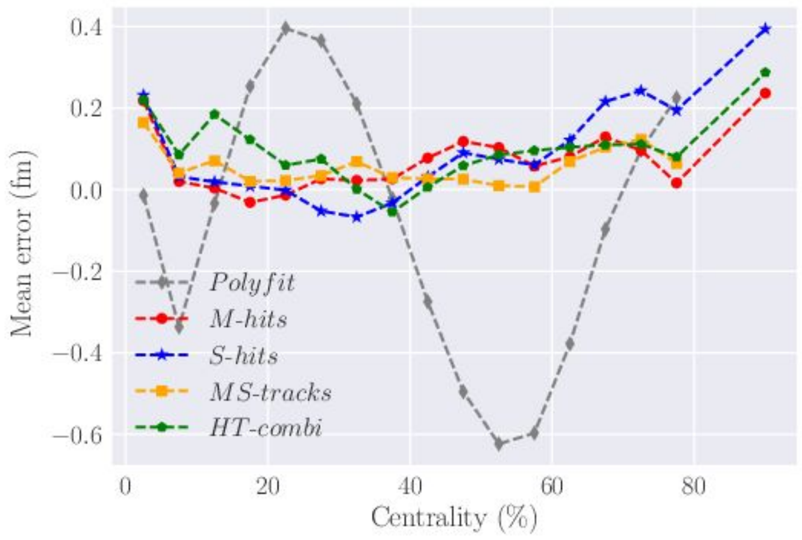
However, the predictions are accurate only if the mean error is close to zero!

Mean error vs. b



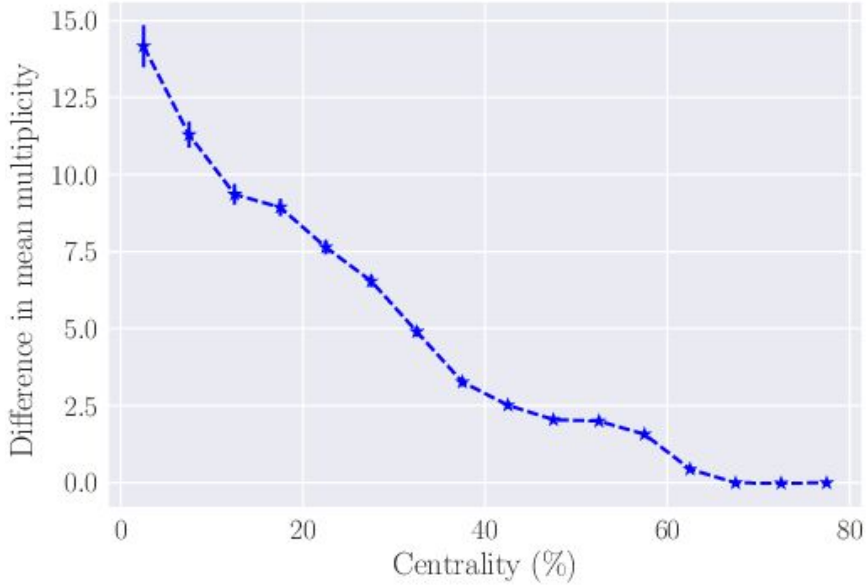
- Quantifies accuracy in predictions
- DL models have mean error between -0.3 and 0.2 fm for b= 2- 14 fm
- Polyfit is highly fluctuating
- All models are less accurate for peripheral events

Mean error vs. centrality



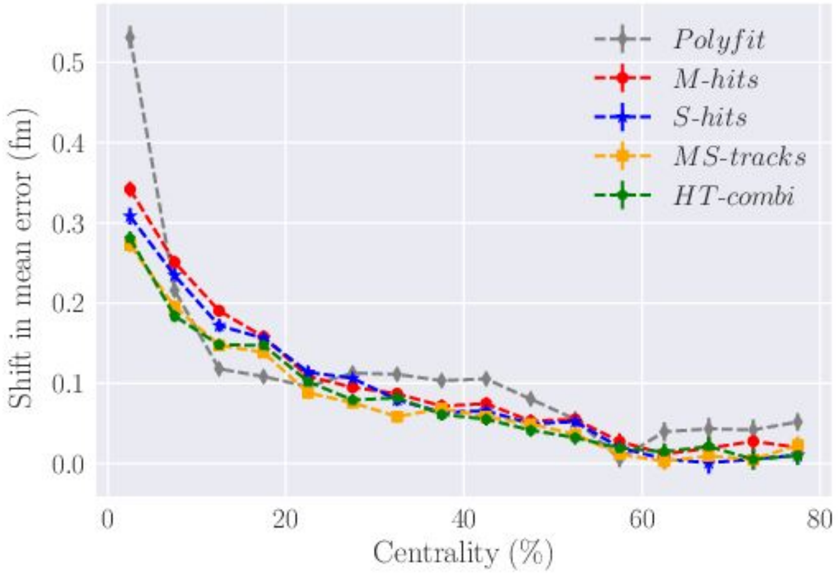
- Data: b=0-16 fm, 1 million events , bdb distribution
- Simulates the realistic b distribution in experiments
- Different from the b distribution of training data
- 5% Centrality bins defined on STS track multiplicity
- DL models have mean error close to zero for most centrality classes

Testing on different physics



- Increased Urqmd pion production cross section
  - Delta baryon absorption decreased by factor of 2
  - Notable for central collisions
- The increased pion production is reflected in mean multiplicity change
  - Difference= New data - Old data

Shift in mean error



- Testing models on modified urqmd data
- Correlation with multiplicity change is visible
- DL models are less dependent than Polyfit especially at central collisions

Testing speed

Model	Speed (events/ s)
M-hits	660
S-hits	159
MS-tracks	1092
HT-combi	435

- Tested on a **Nvidia Geforce RTX 2080 Ti** card with a graphics processing memory of 12 GB
- More room for optimization

Summary

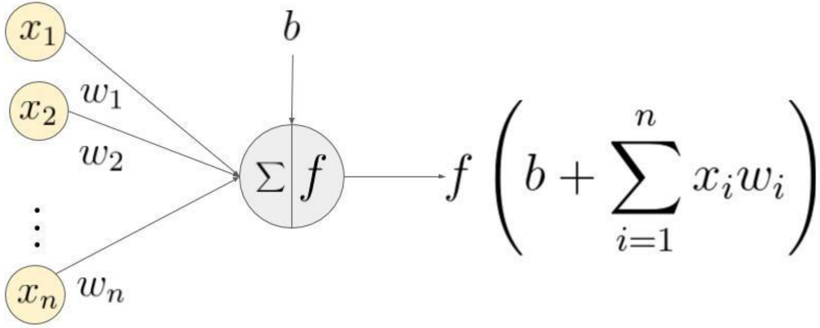
The deep learning models outperforms conventional methods for impact parameter determination

- ❖ Reconstruct the impact parameter on an **event by event** basis
- ❖ Prediction speed upto 1000 events/ s on single GPU
    - ➢ Real time analysis of collected data
- ❖ Reconstruct the impact parameter using the **hit information alone**
    - ➢ Run time AI event selector
    - ➢ Detect faults in detector during data taking
- ❖ Robust to small changes in physics model in comparison to conventional models
- ❖ General framework which can be used for other tasks (e.g. flow extraction)

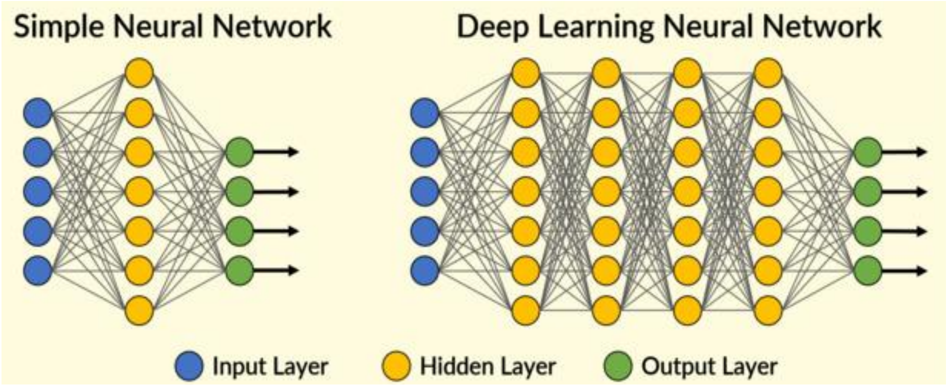
A paper based on this work is available at arXiv:2009.01584

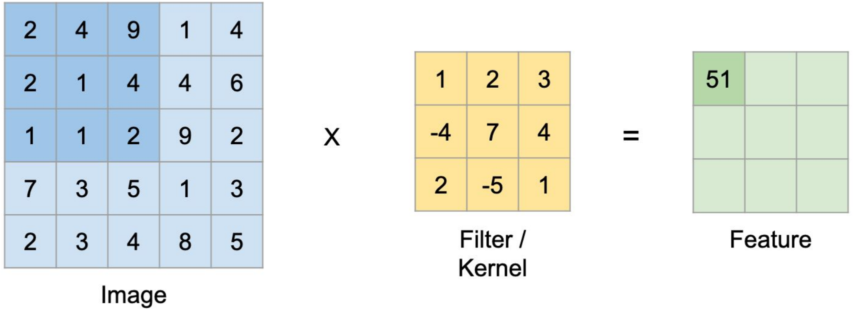
Backup slides

Neural Networks & Convolution Neural Networks

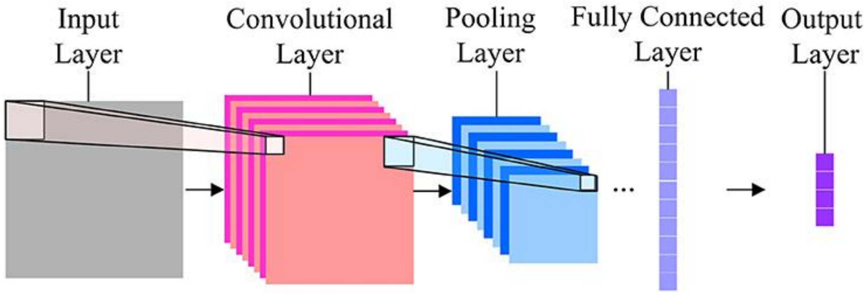


A neuron

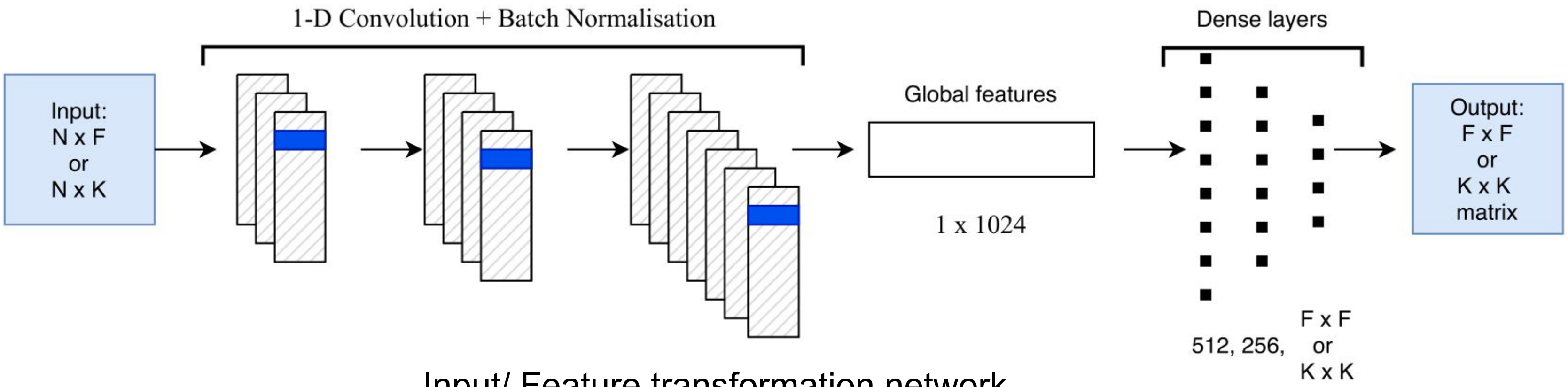




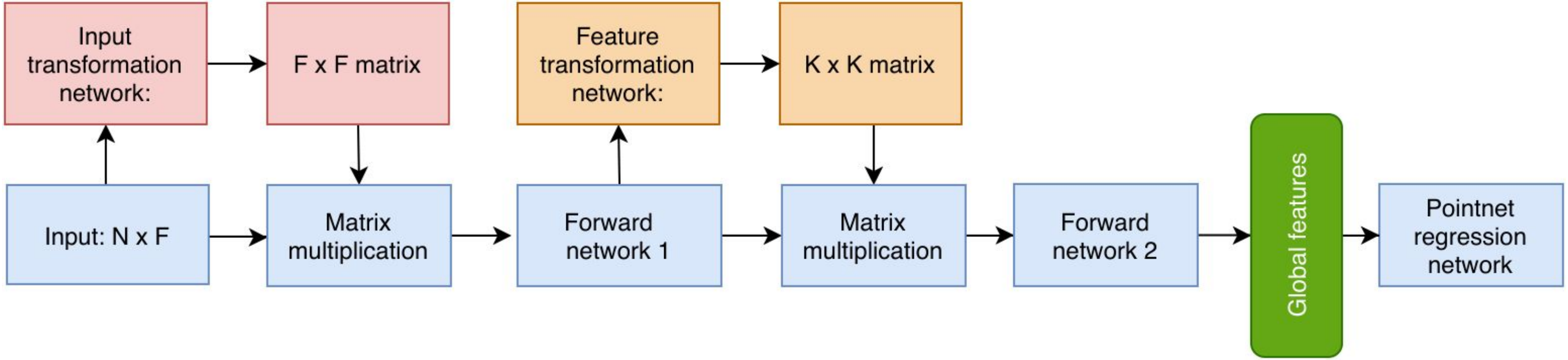
Convolution operation



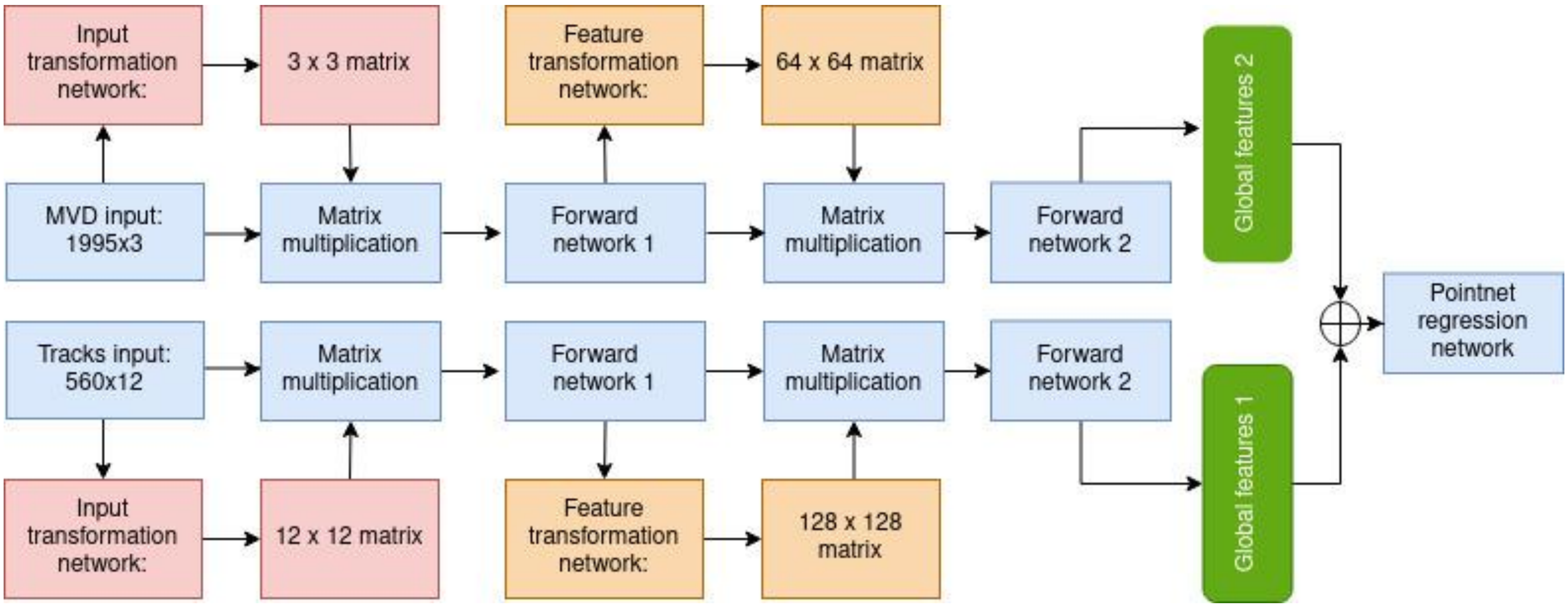
Convolution Neural Network



Input/ Feature transformation network



General structure of Mhits, Shits and MStracks models



Structure of HT-combi model