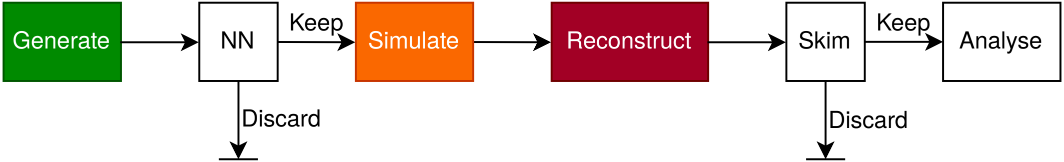
Selective background MC simulation with graph neural networks at Belle II

Nikolai Hartmann[1], James Kahn[2], Yannick Bross[1],
Kilian Lieret[1], Andreas Maximilian Lindner[1], Thomas Kuhr[1]

[1]Ludwig-Maximilians-Universität München (LMU), Germany
[2]Karlsruher Institut für Technologie (KIT), Germany

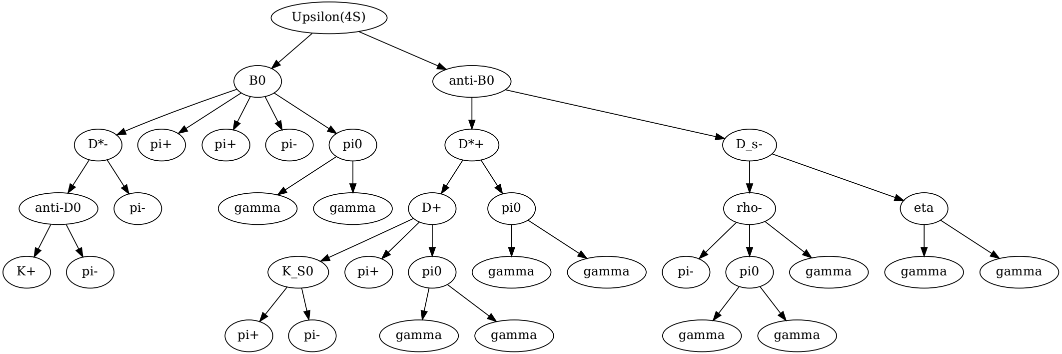
4th IML Machine Learning Workshop, October 23, 2020

The idea

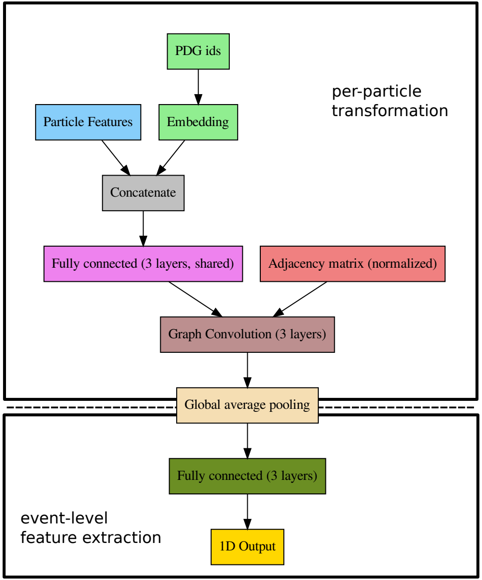


- Event generation takes much less computing time than detector simulation
- Many events discarded (e.g. by skim)
  $\rightarrow$ try to predict which events will be discarded, already after event generation

The MC particle record is a graph (tree)



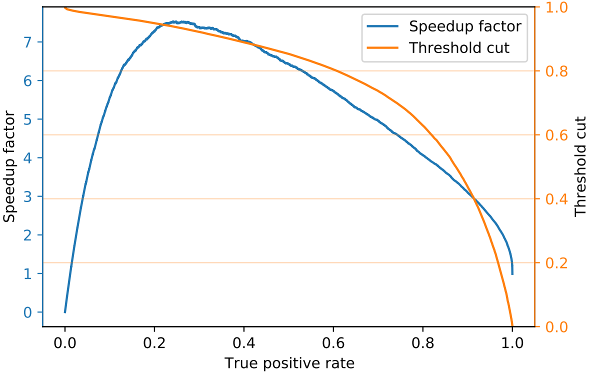
Node attributes: PDG ID, 4-vector components, Vertex positions, Decay times

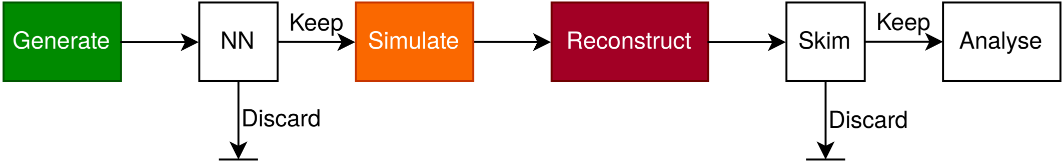

PDG ids
per-particle
transformation
Particle Features
Embedding
Concatenate
Fully connected (3 layers, shared)
Adjacency matrix (normalized)
Graph Convolution (3 layers)
Global average pooling
Fully connected (3 layers)
event-level
feature extraction
1D Output

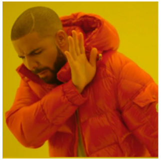
What could we gain?

Hadronic B^0 reconstruction (Full event interpretation skim, 5% retention rate)

Assuming $t_{\mathrm{Simulation+Reconstruction}} = 1000 \times t_{\mathrm{Event\ generation}}$
and $t_{\mathrm{apply\ NN}} = 10 \times t_{\mathrm{Event\ generation}}$



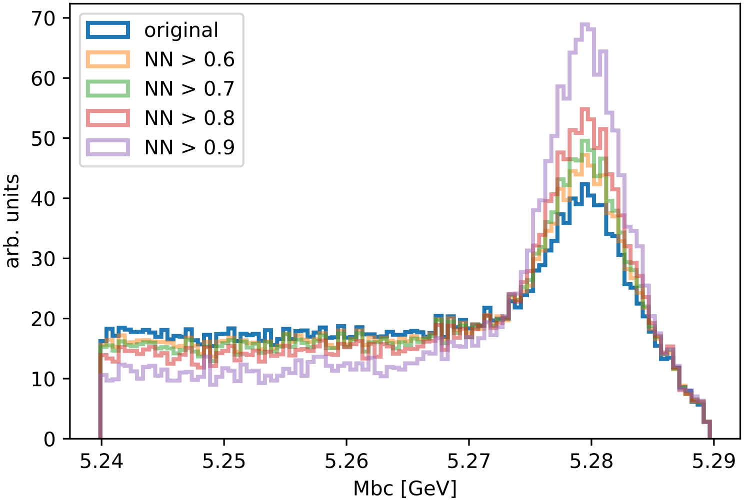




False negatives

False positives

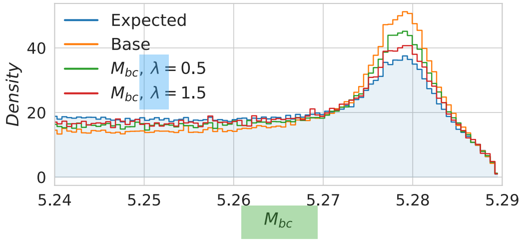
Bias due to false negatives

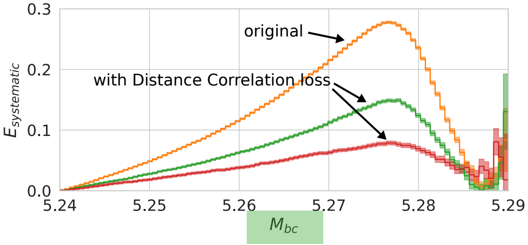


Mitigation via distance correlation loss

Master thesis Yannick Bross

$$L_{\text{tot}} = \text{BCE}(y_{\text{True}}, y_{\text{pred}}) + \lambda \cdot \text{dCorr}(x_{\text{decorr}}, y_{\text{pred}})$$





Same performance (speedup) for all lines!

(Distance correlation: see Wikipedia and arXiv:2001.05310)

How to move on from here?

2 possible Directions:

Bias mitigation + reweighting

- Run Simulation + Reconstruction only for pass events
- Train with bias mitigation
- Do a final “residual reweighting”

“Traditional style” filtering

- Simulate a fraction of discarded events as well
- Use NN output as sampling probability (weight with inverse probability)
- Adjust loss function to account for that

Metric in both cases: Stat. uncertainty on weighted events (for same computing time)

Backup

Simple update rule (Graph convolution, Kipf & Welling)

arXiv:1609.02907

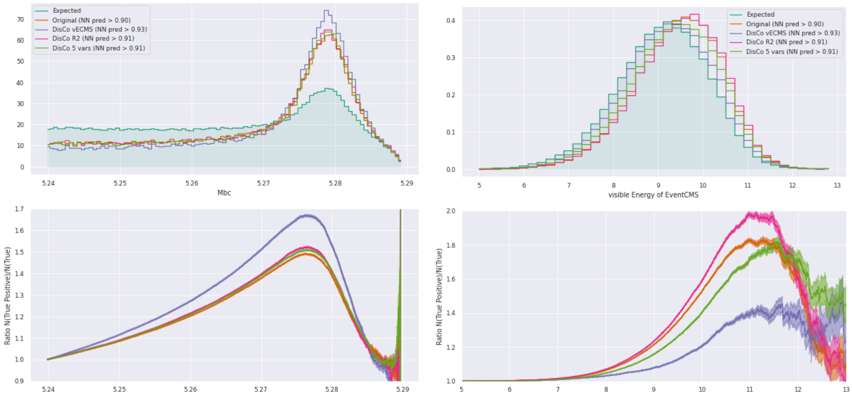
$$\underbrace{\boxed{H^{(l+1)}}}_{\text{New node features}} = \sigma(\underbrace{\boxed{G}}_{\text{Normalized adjacency matrix}}\underbrace{\boxed{H^{l}}}_{\text{Node features (previous layer)}}\underbrace{\boxed{W^{l}}}_{\text{Weights}})$$

Dataset and training

- FEI hadronic B0 skim on mixed samples (Y(4S $\rightarrow$ B0B0bar))
- $\approx$ 1M training events (roughly balanced)
- Particle lists cropped at/padded to 100
  $\rightarrow$ actually works quite well with much less (40 used before)
  $\rightarrow$ mostly crops particles at final stages of decay
- Train with batch size 1024
- Binary cross entropy loss
- Stop after no improvement on validation set (20% of training data, wait 10 epochs)

Effect on other variables

Studies by Yannick



- Bias mitigation is effective for the variable it was trained on
  $\rightarrow$ lower bias for same speedup factor
- mitigation of one quantity can make bias for others worse