Karlsruhe Institute of Technology

Bundesministerium
für Bildung
und Forschung

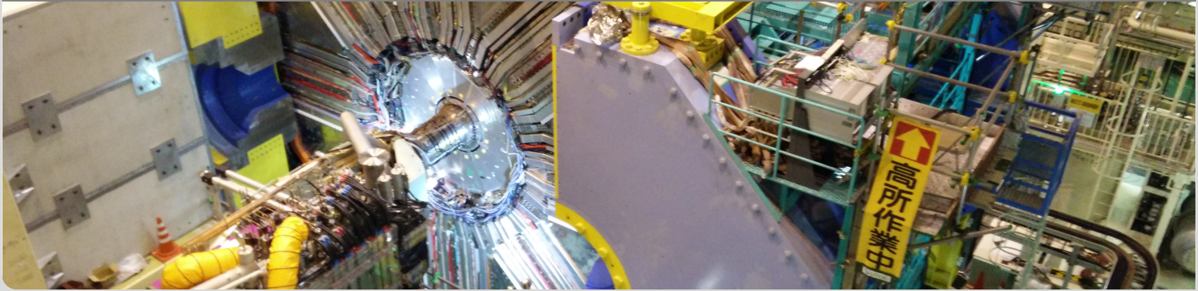
ErUM-Data IDT

Belle II

Selective background Monte Carlo simulation at Belle II

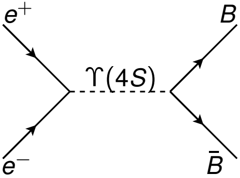
James Kahn, Andreas Lindner, Thomas Kuhr | 5th November 2019

INSTITUT FÜR EXPERIMENTELLE TEILCHENPHYSIK (ETP)

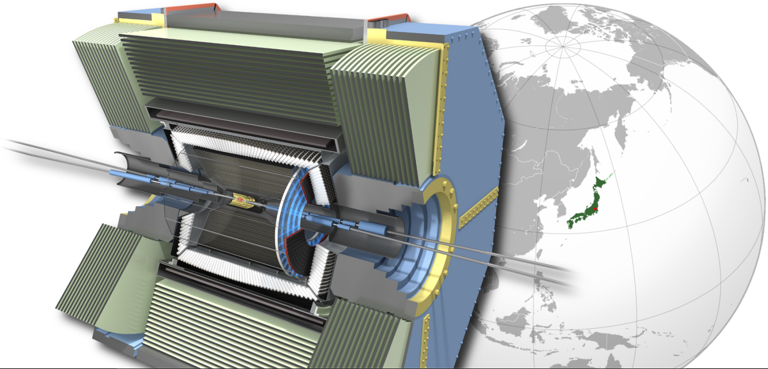


Belle II Experiment

Asymmetric e^+e^- experiment mainly at the $\Upsilon(4S)$ resonance (10.58 GeV)



Focus on B, charm and τ physics



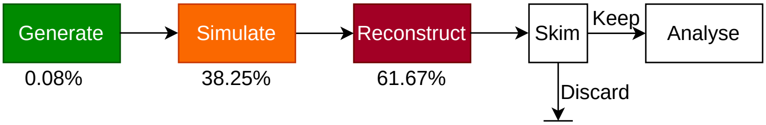
	KEKB/Belle	SuperKEKB/Belle II
Operation	1999–2010	2019–2027
Peak luminosity	$2.11 \times 10^{34}\,\mathrm{cm}^{-2}\mathrm{s}^{-1}$	$8 \times 10^{35}\,\mathrm{cm}^{-2}\mathrm{s}^{-1}$
Integrated luminosity	$1\,\mathrm{ab}^{-1}$ (772 million $B\bar{B}$ pairs)	$50\,\mathrm{ab}^{-1}$

Problem

- Approach at Belle:
  - Background MC $\approx 10 \times$ data
- Infeasible at Belle II $\rightarrow$ still require high statistics
- Currently: $\sim 100\,\mathrm{HS06\,s/event}$
  - $1\,\mathrm{ab}^{-1} \approx 80\,\mathrm{GHS06\,s}$

Skims

- Physics working-group specific datasets (26)
- General selections applied to discard trivial backgrounds
- Retain $\mathcal{O}(0.1\text{–}10\%)$ of full dataset



Problem

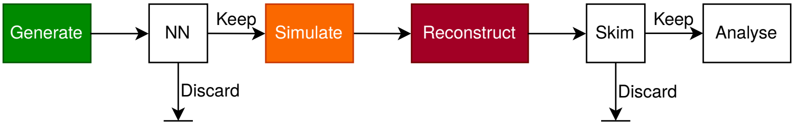
- Approach at Belle:
  - Background MC $\approx 10 \times$ data
- Infeasible at Belle II $\rightarrow$ still require high statistics
- Currently: $\sim 100\,\mathrm{HS06\,s/event}$
  - $1\,\mathrm{ab}^{-1} \approx 80\,\mathrm{GHS06\,s}$

Skims

- Physics working-group specific datasets (26)
- General selections applied to discard trivial backgrounds
- Retain $\mathcal{O}(0.1\text{–}10\%)$ of full dataset

Proposed solution:

Insert NN to predict skims before expensive steps



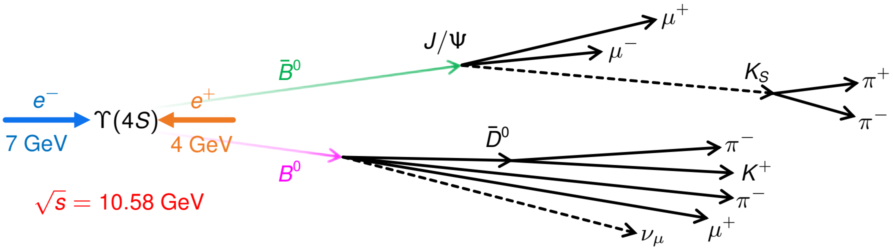
Dataset

$\sim 300,000$ particle collision events with binary classification labels:

- Hadronic $B+$ meson reconstruction ($\sim 5\%$)
- Time-dependent *CP* violation ($\sim 0.2\%$)

Graph terminology

- Nodes = Particles
- Node attributes = Particle properties
- Edges = Parent-daughter relations (decays)
- Graph type = Tree



Dataset

$\sim 300,000$ particle collision events with binary classification labels:

- Hadronic $B+$ meson reconstruction ($\sim 5\%$)
- Time-dependent *CP* violation ($\sim 0.2\%$)

Graph terminology

- Nodes = Particles
- Node attributes = Particle properties
- Edges = Parent-daughter relations (decays)
- Graph type = Tree

$\Upsilon(4S)$ (300553)
  $\bar{B}^0$ (-511)
    J/ψ (443)
      μ^+ (-13)
      μ^- (13)
    K^0_S (310)
      pi^- (-211)
      pi^+ (211)
  B^0 (511)
    $\bar{D}^0$ (-421)
      pi^- (-211)
      K^+ (321)
    pi^- (-211)
    μ^+ (-13)
    ν_μ (14)

Feature	Definition
PDG code	Identifier of particle type and charge.
Mother PDG code	Particle parent PDG code.
Mass	Particle mass in GeV/c^2.
Charge	Electric charge of the particle.
Energy	Particle energy in GeV.
Momentum	Three momentum of the particle in Gev/c.
Production time	Production time in ns relative to $\Upsilon(4S)$ production.
Production vertex	Coordinates of particle production vertex.
Status bit	Bitmask representing MC production conditions.

Graph Isomorphism Network

Node N update rule of layer ℓ (Red $=$ trainable):

$$N^{(\ell+1)} = \mathrm{MLP}^{(\ell)} \left(W_p^{(\ell)} N_p^{(\ell)} + W^{(\ell)} N^{(\ell)} + W_d^{(\ell)} \sum_{\text{daughters}} N_d^{(\ell)} \right)$$

Intuition: **Create representation of node considering its neighbours**

- Custom weights for parent (W_p), node (W), daughters (W_d)
- Independent of daughter ordering
- Normalise adjacency matrix
  - Prevent over-representation in high multiplicity decays

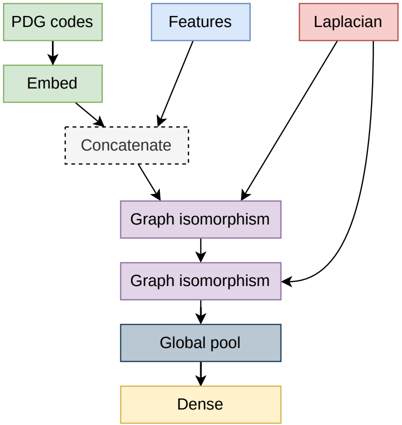
Normalised Laplacian

$$\tilde{A} = A + I_N$$
$$\tilde{D}_{ii} = \sum_j \tilde{A}_{ij}$$
$$\tilde{L} = \tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}}$$

Special case of:
K. Xu, W. Hu, J. Leskovec, S. Jegelka, How Powerful are Graph Neural Networks? (CoRR 2018)

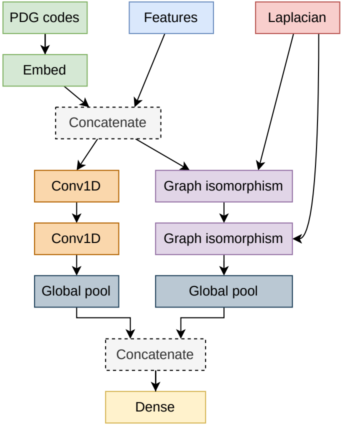
Training

- Train on 250k events (validate on 10%)
- Test on 50k independent events
- Batch normalisation, dropout, class weights, early stopping, reduce LR on plateau, model checkpoint (save only best), . . .



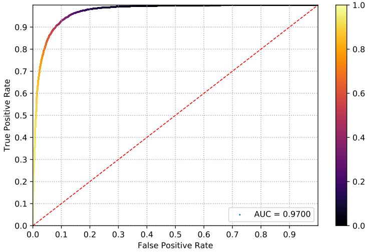
Training

- Train on 250k events (validate on 10%)
- Test on 50k independent events
- Batch normalisation, dropout, class weights, early stopping, reduce LR on plateau, model checkpoint (save only best), ...
- Additional convolutional 1D for full reconstruction dataset

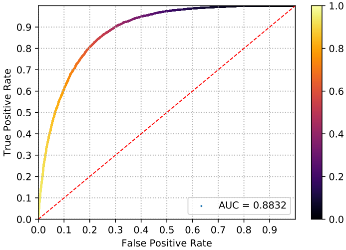


Training

- Train on 250k events (validate on 10%)
- Test on 50k independent events
- Batch normalisation, dropout, class weights, early stopping, reduce LR on plateau, model checkpoint (save only best), . . .
- Additional convolutional 1D for full reconstruction dataset



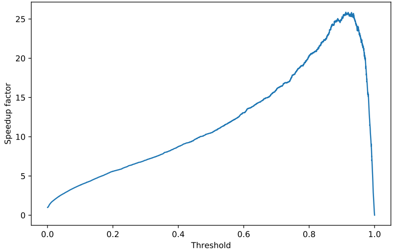
(a) TDCPV



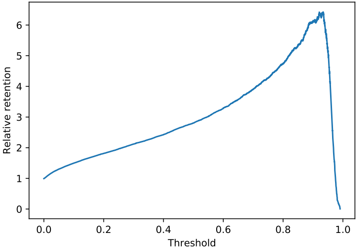
(b) Full reconstruction

Training

- Train on 250k events (validate on 10%)
- Test on 50k independent events
- Batch normalisation, dropout, class weights, early stopping, reduce LR on plateau, model checkpoint (save only best), . . .
- Additional convolutional 1D for full reconstruction dataset
- Insert NumPy-based module into Belle II analysis framework for inference



(a) TDCPV

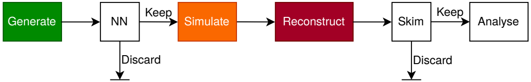


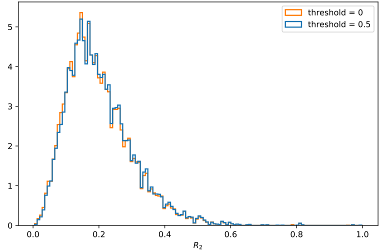
(b) Full reconstruction

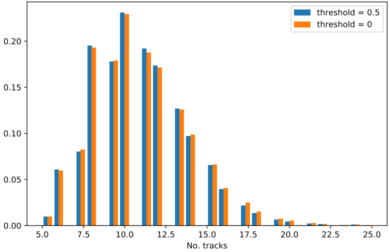
Bias check

Compare event-level kinematics:

- Pass skim = True
- Pass skim and NN = True positive







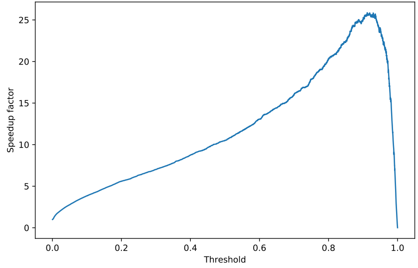
Bias check

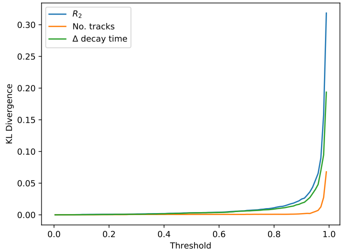
Compare event-level kinematics:

- Pass skim = True
- Pass skim and NN = True positive

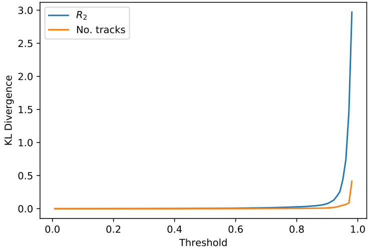
Kullback-Leibler divergence of Q from P:

$$D_{\mathrm{KL}}(P \parallel Q) = -\sum_{x\in\mathcal{X}} P(x)\log\left(\frac{Q(x)}{P(x)}\right)$$





(a) TDCPV



(b) Full reconstruction

Summary

- Belle II has begun data taking
  - simulation will need to keep up
- Simulations for the full 50 ab^{-1} too computationally expensive
  - Requires smarter solutions
- Propose to use NN to go from: **simulate everything** $\rightarrow$ **simulate necessary**
  - Must be general enough to handle each physics working-group case
- Shown potential for orders of magnitude speedup and quantification of bias

Current work:

- Scale up datasets and bias checks
- Implement bias mitigation

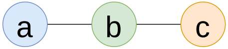
Thank you

Backup

Original Graph Convolutional Networks (GCN)

Propagation rule of layer activations $H^{(l)}$

$$H^{(l+1)} = \sigma\left(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}H^{(l)}W^{(l)}\right)$$

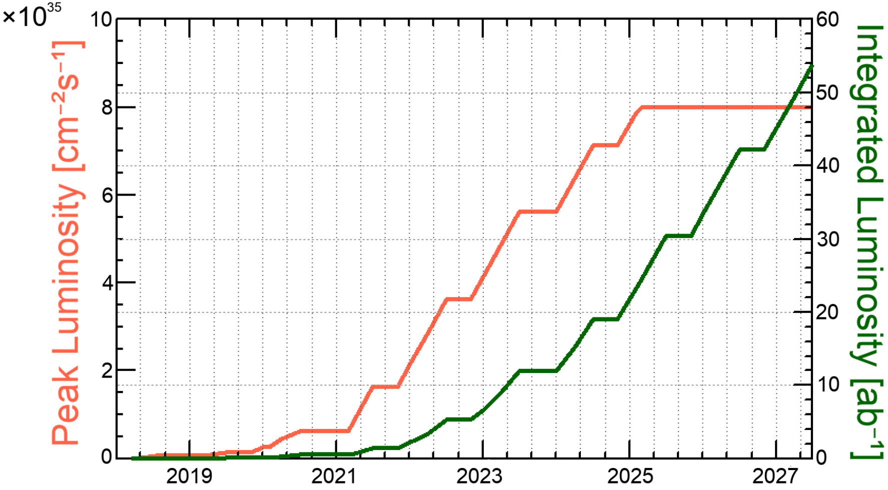


$$H^{(0)} = X$$
$$\tilde{A} = A + I_N$$
$$\tilde{D}_{ii} = \textstyle\sum_j \tilde{A}_{ij}$$

$$\tilde{A}^{N\times N} = A + I = \begin{array}{c c} & \begin{array}{ccc} a & b & c \end{array} \\ \begin{array}{c} a \\ b \\ c \end{array} & \begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 1 \end{bmatrix} \end{array}$$

Thomas N. Kipf, Max Welling, Semi-Supervised Classification with Graph Convolutional Networks (ICLR 2017)

Luminosity projection



TDCPV divergence (overload)