Selective background Monte Carlo simulation at Belle II

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

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

Focus on B, charm and $\tau$ physics

<table>
<thead>
<tr>
<th>KEKB/Belle</th>
<th>SuperKEKB/Belle II</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operation</td>
<td>1999–2010</td>
</tr>
<tr>
<td>Peak luminosity</td>
<td>$2.11 \times 10^{34} \text{ cm}^{-2} \text{s}^{-1}$</td>
</tr>
<tr>
<td>Integrated luminosity</td>
<td>1 ab$^{-1}$ (772 million $B\bar{B}$ pairs)</td>
</tr>
</tbody>
</table>
Problem

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

Skims

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

---

Diagram:

- Generate: 0.08%
- Simulate: 38.25%
- Reconstruct: 61.67%
- Skim
  - Keep: 38.25%
  - Discard: 61.67%
- Analyse
Problem

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

Proposed solution:
Insert NN to predict skims before expensive steps

Skims

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

∼ 300,000 particle collision events with binary classification labels:

- Hadronic $B^+$ meson reconstruction (∼ 5%)
- Time-dependent $CP$ violation (∼ 0.2%)

Graph terminology

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

$\Upsilon(4S)$
$\sqrt{s} = 10.58 \text{ GeV}$
Dataset

∼ 300,000 particle collision events with binary classification labels:
- Hadronic $B^+$ meson reconstruction (∼ 5%)
- Time-dependent $CP$ violation (∼ 0.2%)

Graph terminology

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

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

\[
\begin{align*}
\gamma(4S) & (300553) \\
\bar{B}^0 & (-511) \\
\Upsilon & (443) \\
\mu^+ & (-13) \\
\mu^- & (13) \\
K^0_s & (310) \\
\pi^+ & (211) \\
\pi^- & (-211) \\
D^0 & (511) \\
D^* & (-421) \\
K^0 & (321) \\
\chi & (211) \\
\mu^+ & (-13) \\
\nu & (14)
\end{align*}
\]
Graph Isomorphism Network

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

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

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
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 $P$:

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

(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 $H^{(l)}$

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

$H^{(0)} = X$

$\tilde{A} = A + I_N$

$\tilde{D}_{ii} = \sum_j \tilde{A}_{ij}$

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

Peak Luminosity [cm\(^{-2}\text{s}^{-1}\)]

Integrated Luminosity [ab\(^{-1}\)]

\(\times 10^{35}\)

2019 2021 2023 2025 2027
TDCPV divergence (overload)

KL Divergence

- nTracks
- abs(daughter0.DiffOf(0,1,mcDecayTime))
- cosTheta
- cosThrustOfEvent
- daughter(0,cosTheta)
- Q
- aplanarity
- backwardHemisphereEnergy
- backwardHemisphereMass
- backwardHemisphereMomentum
- backwardHemisphereX
- backwardHemisphereY
- backwardHemisphereZ
- sphericity
- thrust
- cleoConeThrust0
- cleoConeThrust1
- cleoConeThrust2
- cleoConeThrust3
- cleoConeThrust4
- cleoConeThrust5
- cleoConeThrust6
- cleoConeThrust7
- cleoConeThrust8
- foxWolframR1
- foxWolframR2
- foxWolframR3
- foxWolframR4
- missingEnergyOfEventCMS
- missingMass2OfEvent
- missingMomentumOfEvent
- missingMomentumOfEventCMS
- missingMomentumOfEventCMS_Px
- missingMomentumOfEventCMS_Py
- missingMomentumOfEventCMS_Pz
- missingMomentumOfEvent_Px
- missingMomentumOfEvent_Py
- missingMomentumOfEvent_Pz
- missingMomentumOfEvent_theta
- totalPhotonsEnergyOfEvent
- visibleEnergyOfEventCMS

Selective background Monte Carlo simulation at Belle II - James Kahn, Andreas Lindner, Thomas Kuhr 5th November 2019 13/13