Improved $e^+e^- \to \Lambda\bar{\Lambda}$ Selection with Neural Networks

Viktor Thorén, Jacek Biernat

Uppsala University

Workshop of Baryon Production at BESIII, Hefei
2019-09-15

Outline

UPPSALA UNIVERSITET

- **Motivation**
- Neural Network
- Consistency and Systematics Checks
- Prospects
- Summary

ElectroMagnetic Form Factors (EMFFs)

Spin $1/2$ **baryons:**

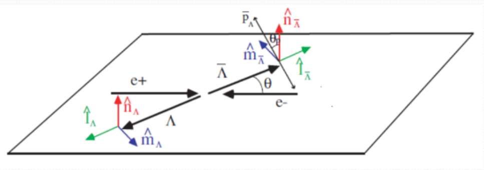
- Two independent EMFFs: $G_E(s)$, $G_M(s)$

  ($s = q^2$, four-momentum transfer)

- In time-like region: Complex, with rel. phase $\Delta\Phi(s)$

$$\frac{d\sigma_{Born}(q^2)}{d\cos\theta} = \frac{2\pi\alpha^2\beta}{4q^2}[(1+\cos^2\theta) + \frac{1}{\tau}R^2(\sin^2\theta)],$$

$$\cos\theta = \hat{p}\cdot\hat{k}$$



Aim of this Study

Study $e^+e^- \to \Lambda\bar{\Lambda}$ at 2.396-2.9 GeV and measure:

- Cross section
- $R = |G_E/G_M|$
- $\Delta\Phi$?

The Method:

- Select events with:
  - Exactly one Λ candidate
  - No $\bar{\Lambda}$ in acceptance.
- Can have three stat. independent samples:
  - Only Λ tagged
  - Only $\bar{\Lambda}$ tagged
  - Both Λ and $\bar{\Lambda}$ tagged (Done at 2.396 arXiv:1903.09421)
- Improve precision by combining samples

Data Sample

Highest statistics points of R-Scan sample:

E_{cm} [GeV]	Integrated Luminosity [pb^{-1}]
2.3864	$22.588 \pm 0.010 \pm 0.174$
2.396	$66.893 \pm 0.017 \pm 0.462$
2.6444	$33.650 \pm 0.013 \pm 0.209$
2.6464	$34.064 \pm 0.013 \pm 0.276$
2.9	$105.53 \pm 0.025 \pm 0.897$

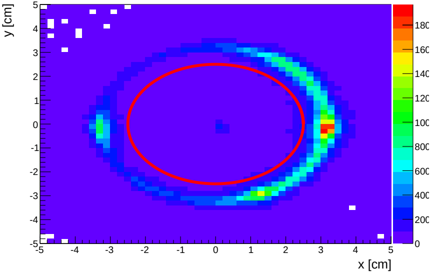
Two points at 2.64 GeV analysed together.

MC samples:

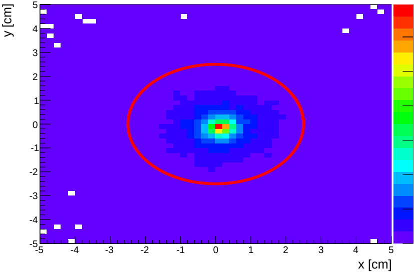
- 50M PHSP at each energy
- 1M ConExc (Angles+ISR) at each energy
- 10M "mDIY" (Angles) at each energy

Beam Pipe Background

- Λ-vertex distribution shows background from beam pipe.
- Background not described by MC
- Naive solution is a vertex cut (red circle)
  - → Risk losing signal. Can one do better?



(a) Data



(b) Signal MC

What can we do?

Typical Procedure:

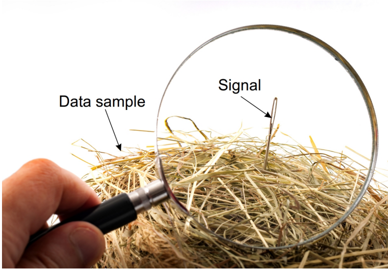
1. Model background
2. Find useful patterns "by eye"

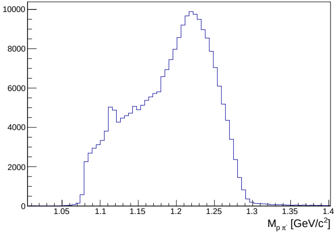
However:

- Background is right there in the data.

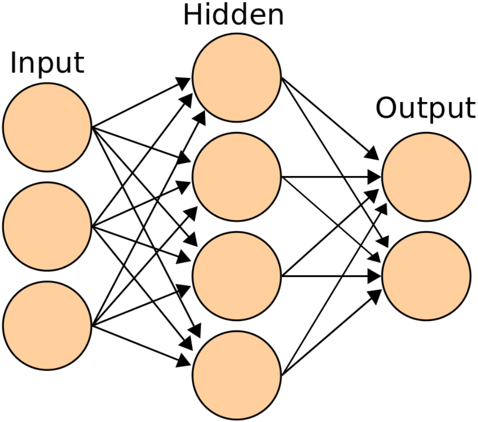
→ Not necessary to model it!

→ Find patterns with machine learning





Artifical Neural Networks



Hidden layer: $H = \mathbf{W}\bar{y} + \bar{b}$,

Activation Function: $f(H) = max(0, H)$

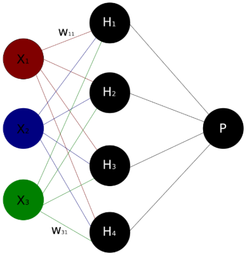
Bayesian Neural Networks

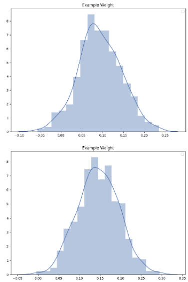
How are they different?

- Approximate posterior distribution of weights given data
  - $\rightarrow$ Weights are distributions, not fixed numbers
  - $\rightarrow$ Sample from distributions to evaluate model

Advantages:

- Ensemble model. Combats overfitting
- Can assign uncertainties to predictions





Neural Network: Implementation, Training and Input

Software: (Originally by project students T. Littau and T. Nordahl)

- TensorFlow with Keras in Python
- pyROOT

Training:

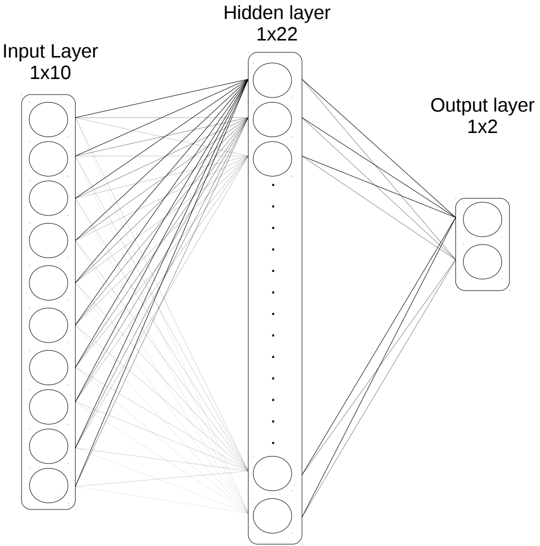
- Signal represented by ConExc MC sample
- Background represented by data.

Architecture: (see also next slide)

- 8 input features

  [Decay Length, LambdaVertexX, LambdaVertexY, LambdaVertexZ, mdcStartpionX, mdcStartpionY, mdcStartpionZ, mdcStartprotonX, mdcStartprotonY, mdcStartprotonZ]
- 1 Hidden Layers, 22 neurons
- Output layer, 2 classes

Network Architecture



Training the Network

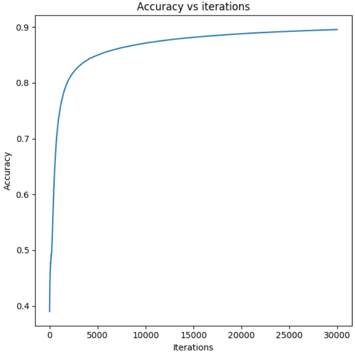
1. Label events: '0' for background. '1' for signal
2. Create training dataset
3. Process data in batches of 100. Average 'loss function' over batch
   - Processing one batch = one **iteration**
   - Reading through the whole dataset one time = one **epoch**
4. Update weights
5. When accuracy on validation data converges, apply on full data set.

Convergence: When is the training done?

$$\text{Acc.} = \frac{f\big(\big(\text{true labels}\big) - \big(\text{predicted labels}\big)\big)}{N_{events}},$$

where

$$f(x) = \begin{cases} 1, & \text{if } x = 0 \\ 0, & \text{else} \end{cases},$$

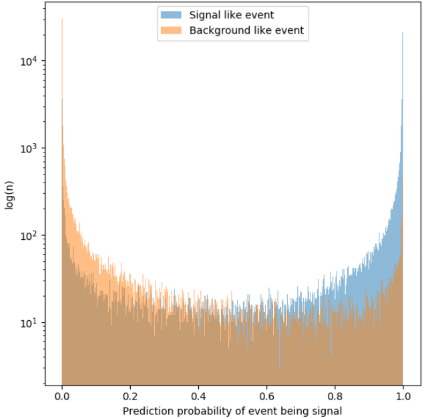


Applying a Trained Network

1. Sample model weights from trained distributions
2. Compute model output
3. Repeat (1) and (2) a large number of times
4. Average over all samples to get final result

Output

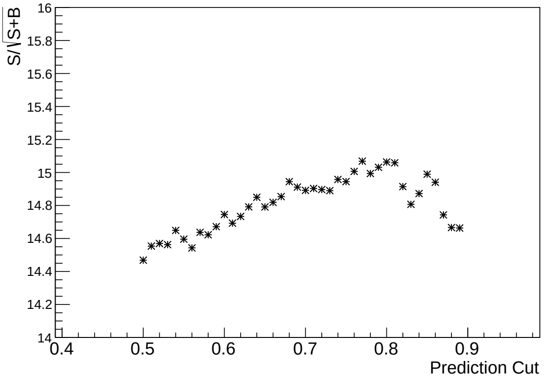
- Degree of belief that event belongs to class 1
- Obtain final sample by cut
  - → Threshold be optimized on case-by-case basis



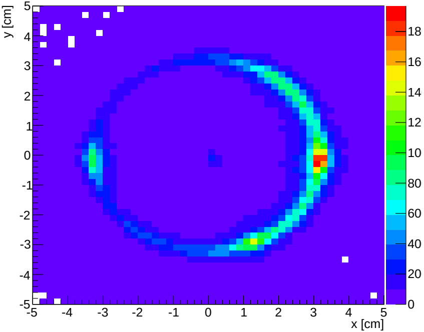
UPPSALA UNIVERSITET

Decision Cut

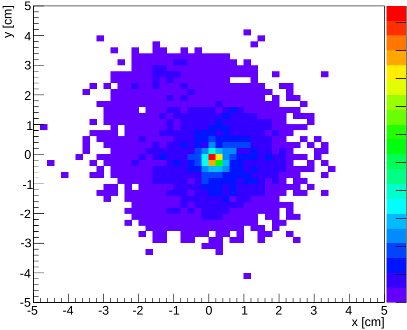
Typical cut optimization plot:



Effect on Beam Pipe Background

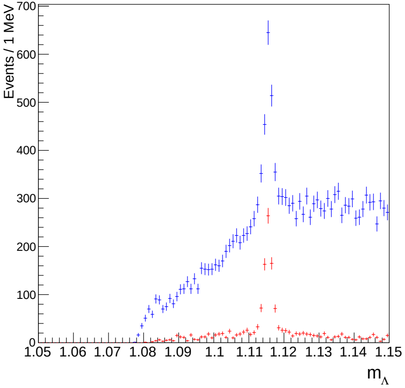


(a) Before



(b) After

Effect on m_Λ



Efficiencies

- Estimated with ConExc MC
- We select two types of signal events:
  - Type 1: $\Lambda \to p\pi^-$, $\bar{\Lambda} \to \bar{p}\pi^+$
  - Type 2: $\Lambda \to p\pi^-$, $\bar{\Lambda} \to \bar{n}\pi^0$
- For brevity we identify:
  - $\epsilon_1 = \epsilon_{\Lambda\to p\pi^-, \bar{\Lambda}\to\bar{p}\pi^+}$
  - $\epsilon_2 = \epsilon_{\Lambda\to p\pi^-, \bar{\Lambda}\to\bar{n}\pi^0}$

Final efficiencies: $\epsilon_1 = 9.8$ %, $\epsilon_2 = 21.8$ %

(From Vertex cut: $\epsilon_1 = 10.5\%$, $\epsilon_2 = 23.6\%$)

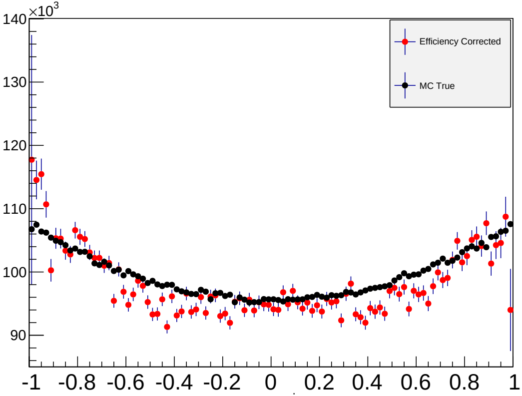
	Vertex Cut	Bayesian Neural Network
Background in sample	34 %	19 %
Signal Efficiency ϵ_1	10.5 %	9.8 %
Signal Efficiency ϵ_2	23.6 %	21.8 %

UPPSALA UNIVERSITET

Efficiency Correction

1. Analyse 50M PHSP events
2. Form 1D histogram of correction factors. 1000 bins.
3. Get factor and correct data event-by-event

Consistency Check: Compare eff. corr. and true MC



UPPSALA UNIVERSITET

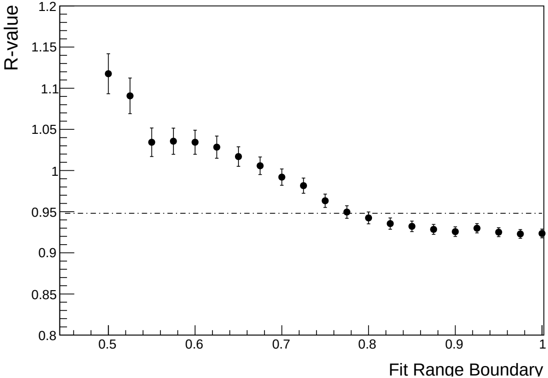
- Motivation
- Neural Network
- **Consistency and Systematics Checks**
- Prospects
- Summary

Input/Output Check

Fit rec. and corr. 10M MC events generated with R=0.948

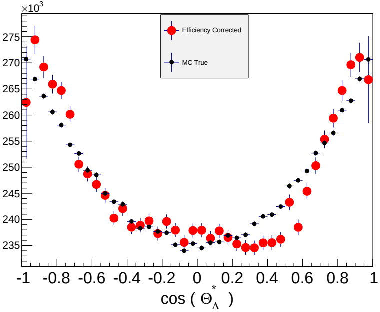
Fit function: $N \times (1 + \cos^2\theta_\Lambda + (\frac{R^2}{\tau})(1 - \cos^2\theta_\Lambda))$

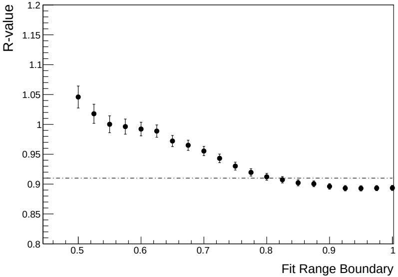
$\tau = \frac{q^2}{4m_\Lambda} = 1.15$



Efficiency Correction: 2.3864

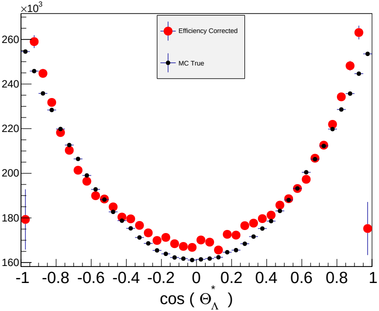
10 MC Events generated with R = 0.91

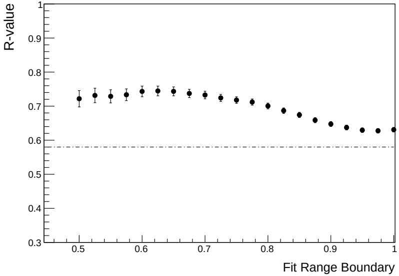




Efficiency Correction: 2.64

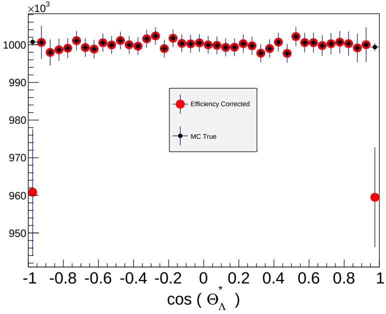
10 MC Events generated with R = 0.58



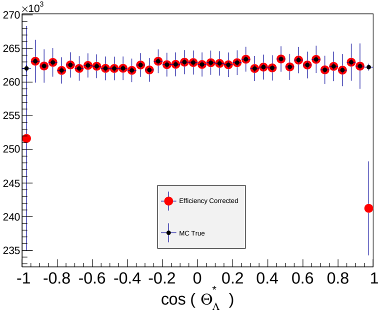


Consistency Checks

PHSP correction factors applied to PHSP sample



2.3864

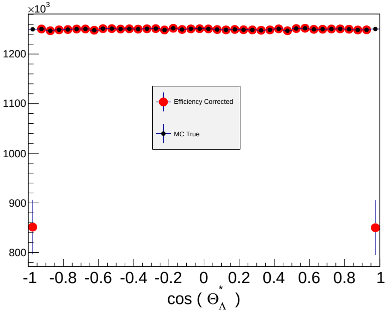


2.396

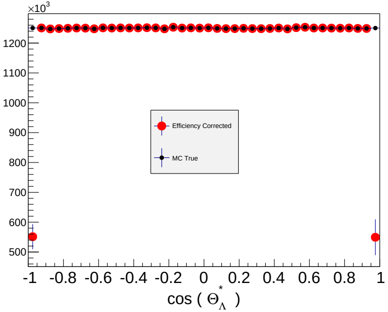
UPPSALA UNIVERSITET

Consistency Checks

PHSP correction factors applied to PHSP sample

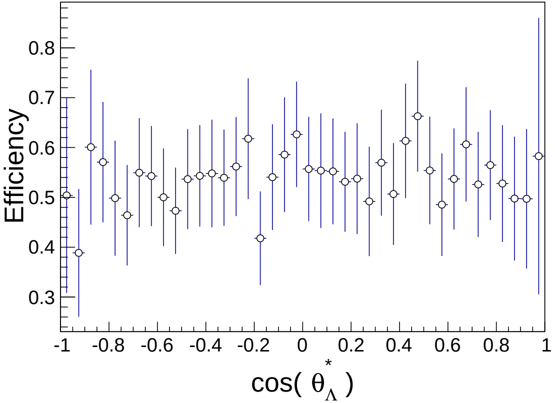


2.64



2.9

Relative Efficiency of NN Cut



UPPSALA UNIVERSITET

Motivation

Neural Network

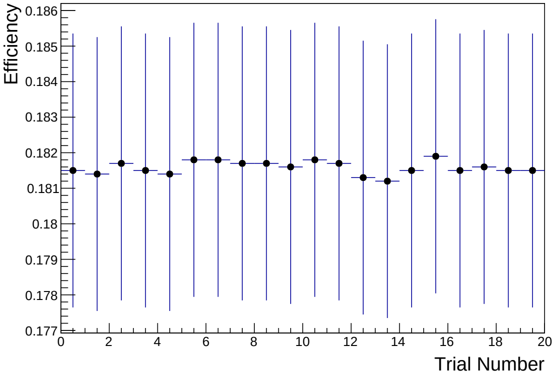
Consistency and Systematics Checks

Prospects

Summary

Check: Effect of Weight Sampling

Apply network 20 times to the same sample



Uppsala Universitet

Future Prospects

How can the method evolve?

- Use more information
  - Add *e.g.* "raw" detector output
- Modify architecture
  - Activation function, loss function, number of nodes, *etc.*
- Constructs new "features"
  - Physics based: *e.g.* Lorentz Boost Network

    arXiv:1812.09722v1 [hep-ex]
  - Arbitrary: *e.g.*
- Use unsupervised algorithms
  - Find anomalies: *e.g.* autoencoders

    arXiv:1812.09722v1 [hep-ex]
  - Identify Patterns: *e.g.* Self-organizing maps

Summary & Outlook

Summary

- BNN improves signal to background ratio over vertex cut
- Edge effects limit fit range for R-determination
- New systematics arise from BNN method
  - Effect from no. of training iterations, weight sampling is small
  - Impact of decision cut non-negligible

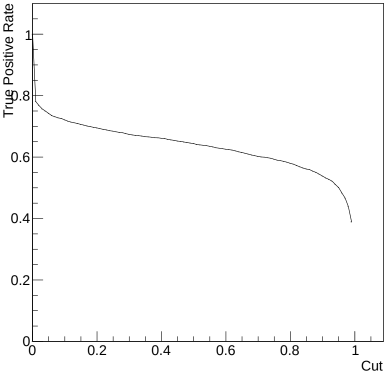
Outlook

- Systematics to be quantified, *e.g.* due to bias in fit
- Software is straightforward to use and can be applied to other reactions with minimal modifications

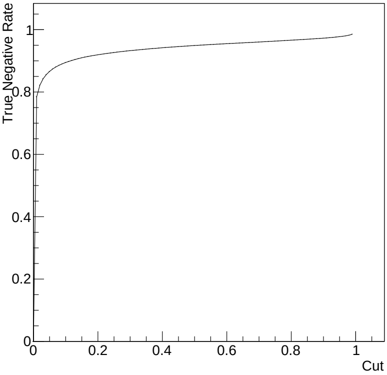
UPPSALA UNIVERSITET

- Motivation
- Neural Network
- Consistency and Systematics Checks
- Prospects
- **Summary**

Backup: Decision Cut

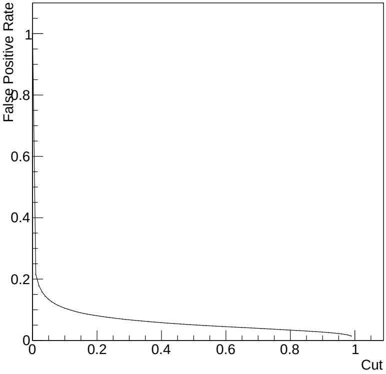


True Positive Rate

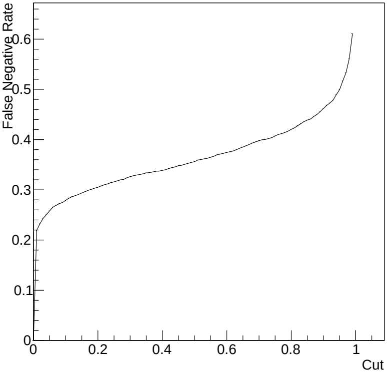


True Negative Rate

Backup: Decision Cut

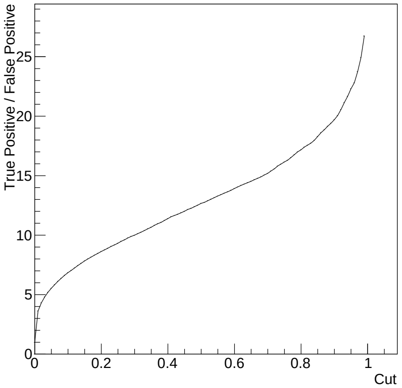


False Positive Rate



False Negative Rate

Backup: Decision Cut



True Positive rate/False Positive rate

UPPSALA UNIVERSITET

Backup: Cut Flow (2.3864)

Cut	Signal Remaining Ch. 1 [%]	Signal Remaining Ch 2. [%
Decay Length	15.79 ± 0.04	32.89 ± 0.04
Momentum window	13.73 ± 0.03	29.09 ± 0.03
Inv. Mass window	12.75 ± 0.03	27.10 ± 0.03
NN. Prediction ≥ 0.95	7.81 ± 0.03	19.96 ± 0.03

UPPSALA UNIVERSITET

Backup:Cut Flow (2.64)

Cut	Signal Remaining Ch. 1 [%]	Signal Remaining Ch 2. [%
Decay Length	11.38 ± 0.04	22.7 ± 0.04
Momentum window	10.62 ± 0.03	21.45 ± 0.03
Inv. Mass window	9.83 ± 0.03	19.94 ± 0.03
NN. Prediction ≥ 0.95	4.20 ± 0.03	12.58 ± 0.03

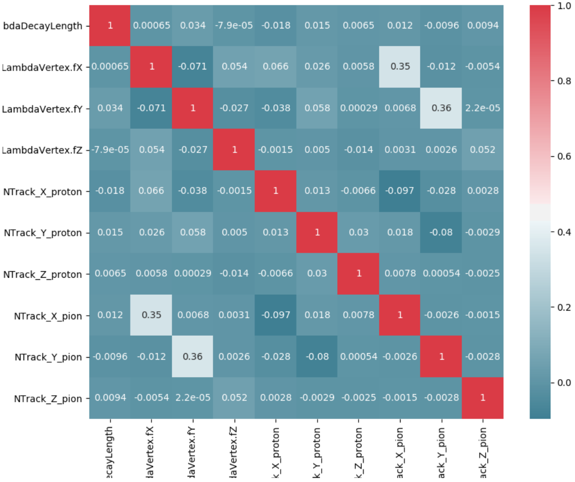
Backup:Cut Flow (2.9)

Cut	Signal Remaining Ch. 1 [%]	Signal Remaining Ch 2. [%
Decay Length	10.00 ± 0.03	18.78 ± 0.04
Momentum window	9.33 ± 0.03	17.71 ± 0.04
Inv. Mass window	8.67 ± 0.03	16.52 ± 0.04
NN. Prediction ≥ 0.71	5.79 ± 0.02	14.74 ± 0.04

Backup: All Efficiencies

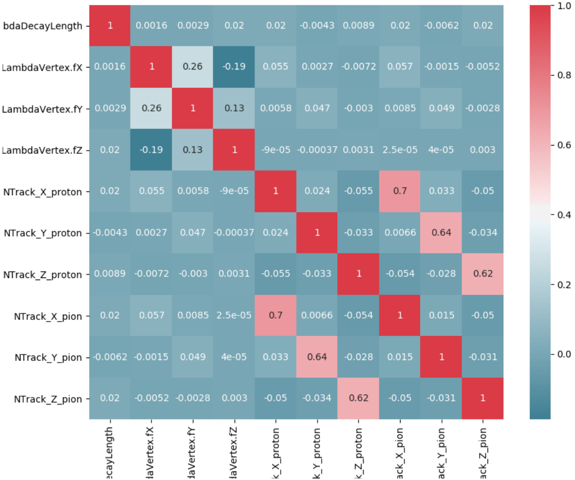
Energy [GeV]	ϵ_1 [%]	ϵ_2 [%]
2.3864	7.81 ± 0.03	19.96 ± 0.03
2.396	9.8 ± 0.03	21.8 ± 0.04
2.64	4.2 ± 0.03	12.58 ± 0.03
2.9	5.79 ± 0.02	14.74 ± 0.04

Backup: Signal Correlations



UPPSALA UNIVERSITET

Backup: Background Correlations



Backup: Comparison with Vertex Cut

Comparable Efficiencies:

- $\epsilon_1 = 10.5\%$, $\epsilon_2 = 23.6\%$ from Vertex cut
- $\epsilon_1 = 10.\%$, $\epsilon_2 = 23.6\%$ from Vertex cut

But lowered background:

- 34 % After vertex cut
- 20 % After NN