Suppression of Continuum Background with Neural Networks for Belle II

Bela Urbschat

Max Planck Institute for Physics, Technical University of Munich

December 19, 2023

2023-12-19

Suppression of Continuum Background with Neural Networks for Belle II

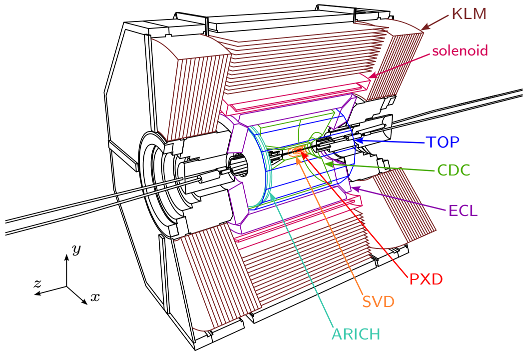
Bela Urbschat

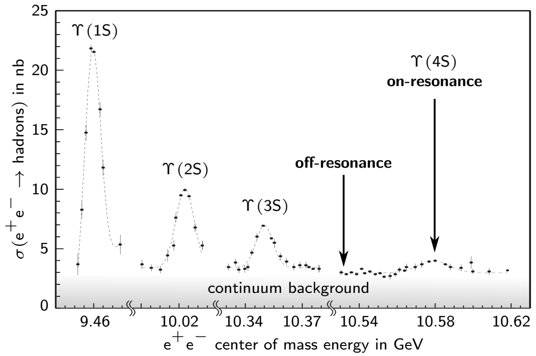
Max Planck Institute for Physics, Technical University of Munich

December 19, 2023

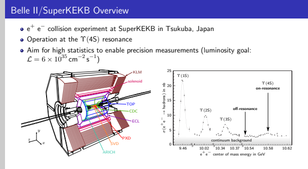
Belle II/SuperKEKB Overview

- $e^+ e^-$ collision experiment at SuperKEKB in Tsukuba, Japan
- Operation at the $\Upsilon(4S)$ resonance
- Aim for high statistics to enable precision measurements (luminosity goal: $\mathcal{L} = 6 \times 10^{35}\,\mathrm{cm}^{-2}\,\mathrm{s}^{-1}$)





CS with NNs for Belle II
2023-12-19
Belle II/SuperKEKB Overview

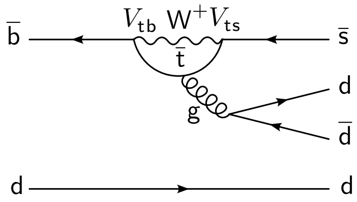


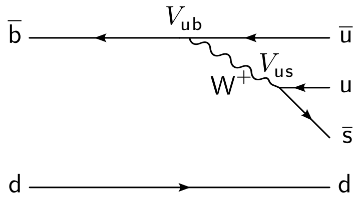
Theoretical Motivation

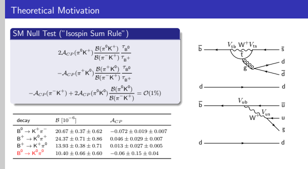
SM Null Test ("Isospin Sum Rule")

$$2\mathcal{A}_{CP}(\pi^0\mathrm{K}^+)\frac{\mathcal{B}(\pi^0\mathrm{K}^+)}{\mathcal{B}(\pi^-\mathrm{K}^+)}\frac{\tau_{\mathrm{B}^0}}{\tau_{\mathrm{B}^+}}$$
$$-\mathcal{A}_{CP}(\pi^+\mathrm{K}^0)\frac{\mathcal{B}(\pi^+\mathrm{K}^0)}{\mathcal{B}(\pi^-\mathrm{K}^+)}\frac{\tau_{\mathrm{B}^0}}{\tau_{\mathrm{B}^+}}$$
$$-\mathcal{A}_{CP}(\pi^-\mathrm{K}^+)+2\mathcal{A}_{CP}(\pi^0\mathrm{K}^0)\frac{\mathcal{B}(\pi^0\mathrm{K}^0)}{\mathcal{B}(\pi^-\mathrm{K}^+)}=\mathcal{O}(1\%)$$

decay	$\mathcal{B}$ $[10^{-6}]$	$\mathcal{A}_{CP}$
$\mathrm{B}^0\to\mathrm{K}^+\pi^-$	$20.67\pm0.37\pm0.62$	$-0.072\pm0.019\pm0.007$
$\mathrm{B}^+\to\mathrm{K}^0\pi^+$	$24.37\pm0.71\pm0.86$	$0.046\pm0.029\pm0.007$
$\mathrm{B}^+\to\mathrm{K}^+\pi^0$	$13.93\pm0.38\pm0.71$	$0.013\pm0.027\pm0.005$
$\mathrm{B}^0\to\mathrm{K}^0\pi^0$	$10.40\pm0.66\pm0.60$	$-0.06\pm0.15\pm0.04$

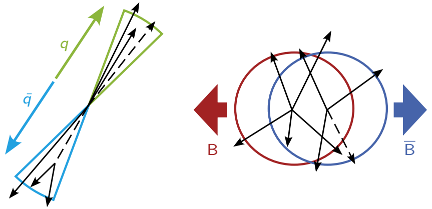


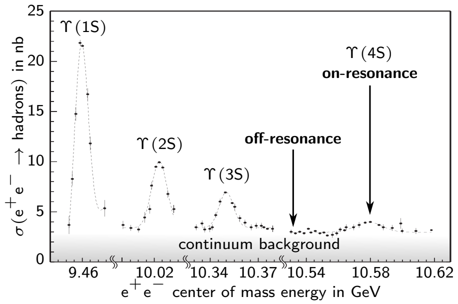




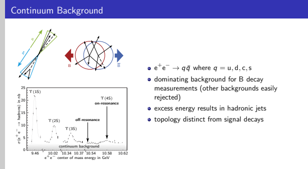
1. Sum rule as null-test to the SM.
2. Holds in isospin symmetry limit (equal quark masses) (right?)
3. Not exactly $=0$, but expected deviation from zero is still much smaller then experimental uncertainties.
4. Highlight the $\mathrm{B}\to\mathrm{K}\pi$ decay modes appearing in sum rule.
5. Highlight that $\mathrm{B}^0\to\mathrm{K}^0\pi^0$ is measured worst (also as not self tagging)
6. NP (particles) could contribute to loops.

Continuum Background





- $\mathrm{e}^+\mathrm{e}^- \to q\bar{q}$ where $q = \mathrm{u, d, c, s}$
- dominating background for B decay measurements (other backgrounds easily rejected)
- excess energy results in hadronic jets
- topology distinct from signal decays



1. Point to the event shape figure.
2. Explain uniform $q\bar{q}$ background in resonances figure.

Continuum Suppression

General Idea

Use topological differences to classify signal and background $\rightarrow$ *thrust frames*

Usual Approach

- Variables *engineered* for continuum suppression
- BDT for classification

Proposed Approach

- Low level **momentum** and **decay vertex** variables
- Attempt to use DNNs, expecting them to excel in extraction of information from low level variables

Past research: Common CS variables augmented with low level variables. Never low level variables exclusively.

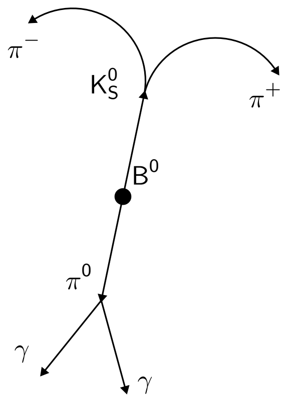
Continuum Suppression
General Idea
Use topological differences to classify signal and background → thrust frames
Usual Approach
- Variables engineered for continuum suppression
- BDT for classification
Proposed Approach
- Low level momentum and decay vertex variables
- Attempt to use DNNs, expecting them to excel in extraction of information from low level variables
Past research: Common CS variables augmented with low level variables. Never low level variables exclusively.

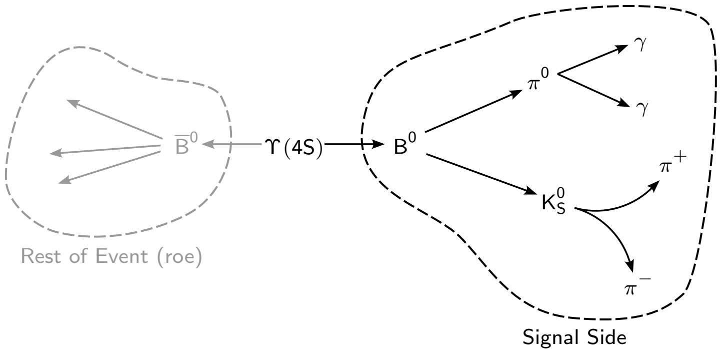
1. Make sure to explain thrust frames!
2. Momentum/vertex variables in theory should contain all the information of event shape.

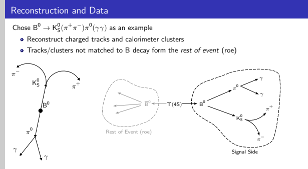
Reconstruction and Data

Chose $B^0 \to K^0_S(\pi^+\pi^-)\pi^0(\gamma\gamma)$ as an example

- Reconstruct charged tracks and calorimeter clusters
- Tracks/clusters not matched to B decay form the *rest of event* (roe)





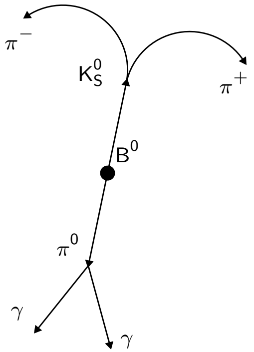


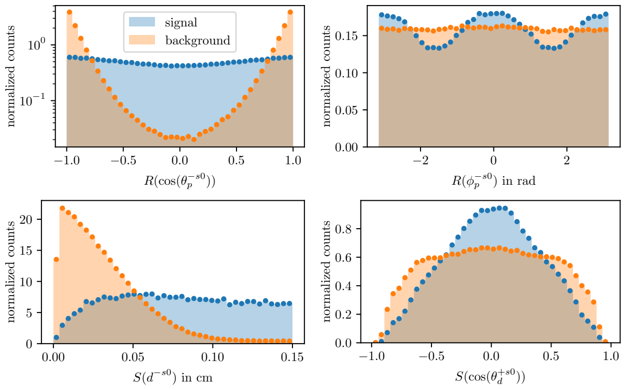
1. Explain signal thrust/roe thrust using figure on the right

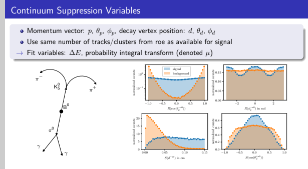
Continuum Suppression Variables

- Momentum vector: p, θ_p, ϕ_p, decay vertex position: d, θ_d, ϕ_d
- Use same number of tracks/clusters from roe as available for signal

$\rightarrow$ Fit variables: ΔE, probability integral transform (denoted μ)







1. Note that we attempted to use more variables from roe which did not result in a significant performance gain
2. Explain chosen orders tracks/clusters for variables
3. Explain notation (briefly)
4. Explain variables that do not fall under the naming scheme
5. Explain intuition for polar angle distribution based on antiparallel/random alignment of thrust axes.

Classifiers Used

Boosted Decision Trees (BTDs)

- Robust classifiers
- Give good baseline for expected performance
- Here no in-depth hyperparameter tuning

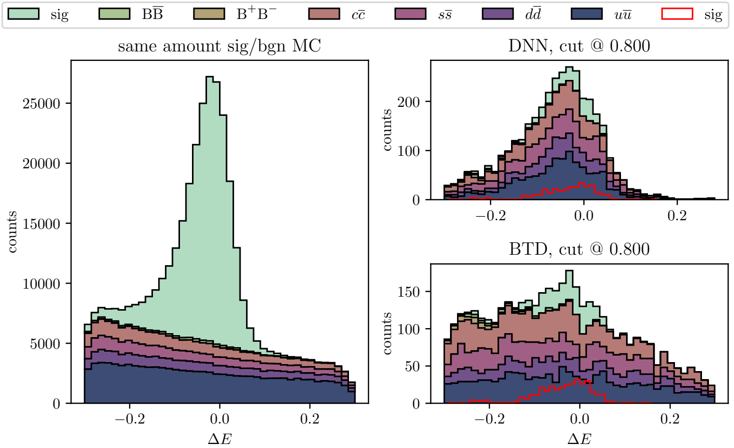
Deep Neural Networks (DNNs)

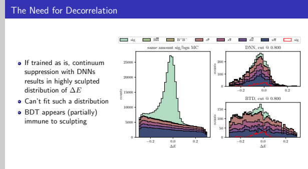
- Initial motivation: Possibly better at utilizing information from low level variables $\rightarrow$ better performance?
- Turn out to be much more delicate/difficult to handle
- Main subject of studies for this thesis

Classifiers Used
Boosted Decision Trees (BTDs)
Robust classifiers
Give good baseline for expected performance
Here no in-depth hyperparameter tuning
Deep Neural Networks (DNNs)
Initial motivation: Possibly better at utilizing information from low level variables → better performance?
Turn out to be much more delicate/difficult to handle
Main subject of studies for this thesis

The Need for Decorrelation

- If trained as is, continuum suppression with DNNs results in highly sculpted distribution of ΔE
- Can't fit such a distribution
- BDT appears (partially) immune to sculpting





1. Explain expected shape using left plot.
2. Highlight that fit with observed level of sculpting is clearly impossible.

Tools(s) for Decorrelation

Distance Correlation

- Efficiently estimable correlation metric, capturing also non-linear correlations
- Only one further hyperparameter introduced

Total loss:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{classifier}}(\vec{y}, \vec{y}_{\text{true}}) + \lambda \cdot \text{dCorr}(\vec{z}, \vec{y})$$

However tuning still difficult:

- Too large λ degrades performance
- Effectiveness of decorrelation also influenced by other hyperparameters (batch size, network architecture)
- Systematic tuning extremely difficult due to conflicting objectives

$\rightarrow$ Studies with preliminary hyperparameters to better understand behavior

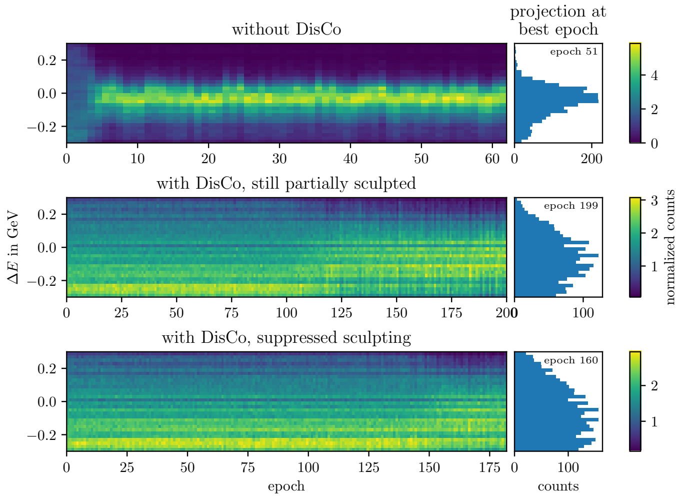
Tools(s) for Decorrelation
Distance Correlation
- Efficiently estimable correlation metric, capturing also non-linear correlations
- Only one further hyperparameter introduced
Total loss:
$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{classifier}}(\vec{y}, \vec{y}_{\text{true}}) + \lambda \cdot \text{dCorr}(\vec{z}, \vec{y})$
However tuning still difficult:
- Too large λ degrades performance
- Effectiveness of decorrelation also influenced by other hyperparameters (batch size, network architecture)
- Systematic tuning extremely difficult due to conflicting objectives
$\rightarrow$ Studies with preliminary hyperparameters to better understand behavior

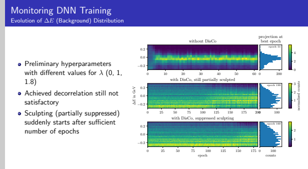
1. Also mention that adversary networks have been implemented, but could not be sufficiently tuned for this thesis.
2. Explain symbols in the equation!
3. Mention that classifier loss is binary cross-entropy.
4. Explain the conflicting objectives of best performance and effective decorrelation (problem: performance always better for correlated classifier).

Monitoring DNN Training

Evolution of ΔE (Background) Distribution

- Preliminary hyperparameters with different values for λ (0, 1, 1.8)
- Achieved decorrelation still not satisfactory
- Sculpting (partially suppressed) suddenly starts after sufficient number of epochs





1. Highlight that after sufficient training (or epochs), correlation (more or less suddenly) starts $\rightarrow$ decorrelation is unstable.
2. Mention that here the goal was to reach lower sculpting than BDT in hope of this improving fit quality (i.e. lowering the statistical uncertainties). Thus the best decorrelation is still not satisfactory.
3. Distributions are normalized at each epoch!

Choice of Hyperparameters

	prelim. value	final value	description
n_{layers}	5	5	number of layers
$n_{\text{neurons},0}$	100	100	1st dense layer neurons
$n_{\text{neurons},1}$	100	100	2nd dense layer neurons
$n_{\text{neurons},2}$	4	6	3rd dense layer neurons
$n_{\text{neurons},3}$	100	100	4th dense layer neurons
$n_{\text{neurons},4}$	100	100	5th dense layer neurons
weight decay	0.000142	0.000142	Weight decay for AdamW
learning rate	0.002	0.015	learning rate
dCorr on bgn	True	True	choice to compute dCorr on only background events
λ	1.8	2	scale of dCorr in total loss
s_λ	7.5	7.5	scale factor for λ when dCorr computed on bgn only
batch size	2048	16384	number of events in a minibatch

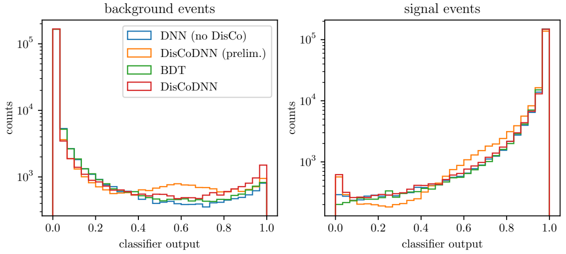
$\rightarrow$ In the following DNN with applied decorrelation and final hyperparameters is referred to as *DisCoDNN*

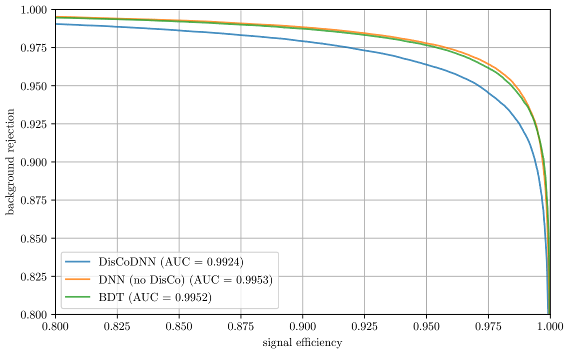
1. Highlight the "unusual" hyperparameters: Large batch size, bottleneck architecture

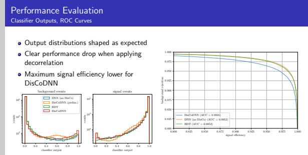
Performance Evaluation

Classifier Outputs, ROC Curves

- Output distributions shaped as expected
- Clear performance drop when applying decorrelation
- Maximum signal efficiency lower for DisCoDNN







1. Note that prelim. DisCoDNN only shown as reference for *not good* output distribution.

ΔE and μ after Continuum Suppression

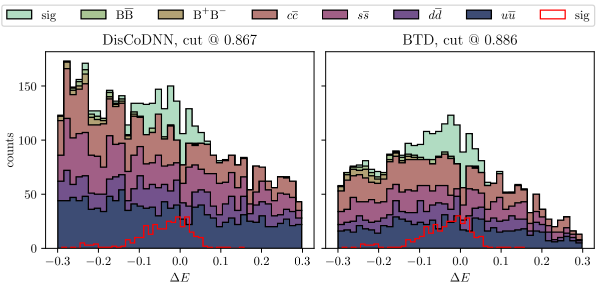
Cuts always chosen for 90% signal efficiency

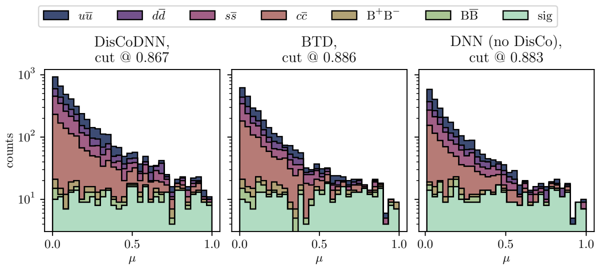
ΔE:

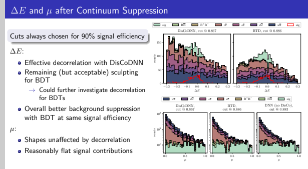
- Effective decorrelation with DisCoDNN
- Remaining (but acceptable) sculpting for BDT
  - $\rightarrow$ Could further investigate decorrelation for BDTs
- Overall better background suppression with BDT at same signal efficiency

μ:

- Shapes unaffected by decorrelation
- Reasonably flat signal contributions

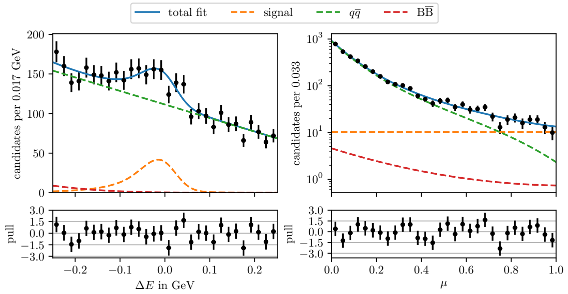




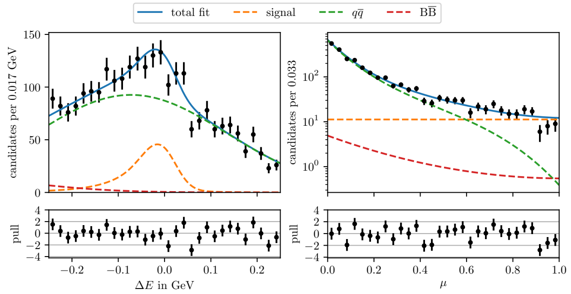


1. Maybe mention how cut positions were determined/that they were determined using an appropriate procedure.

Fits on MC

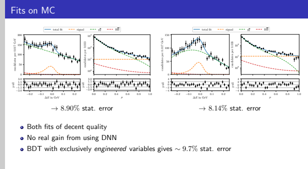


$\rightarrow 8.90\%$ stat. error



$\rightarrow 8.14\%$ stat. error

- Both fits of decent quality
- No real gain from using DNN
- BDT with exclusively *engineered* variables gives $\sim 9.7\%$ stat. error

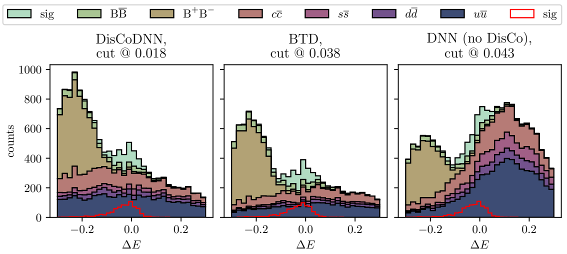


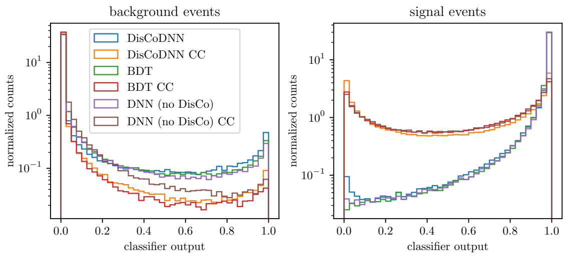
1. Shapes fixed on MC, final fit of only yields

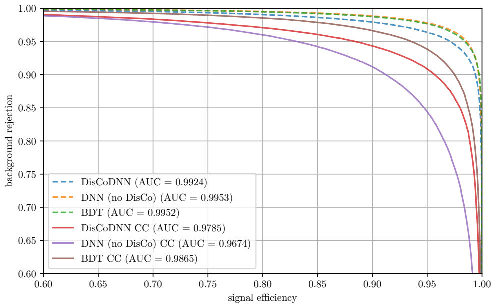
Classifier Generalizability

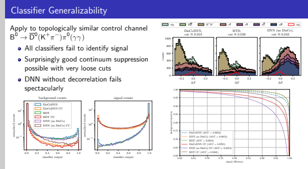
Apply to topologically similar control channel $B^0 \to \overline{D}^0(K^+\pi^-)\pi^0(\gamma\gamma)$

- All classifiers fail to identify signal
- Surprisingly good continuum suppression possible with very loose cuts
- DNN without decorrelation fails spectacularly









1. Mention that this demonstrates the problem of generalizability!
2. Note that DNN (no DisCo) seems *not* to just "compute" or estimate ΔE and then more or less cut on that, as $B\overline{B}$ background remains!
3. Possibly the correlations are then what allows the DNN to sculpt ΔE. This would make sense as DisCoDNN does not really rely on correlations.

Conclusion & Outlook

- Introduced set of low level continuum suppression variables
- Prepared BDT and DNNs using introduced variables, expecting DNN to profit from those
- DNNs require decorrelation, which most likely limits their performance
- Fits on MC show similar accuracies for BDT/DNN but slighly better than BDT with common CS variables

→ Low level CS variables could reduce statistical errors but further investigation (e.g. systematics etc.) needed for final judgement

For the Future

- Study influence of single variables on sculpting (to possibly exclude them)
- Impact on performance with alternative decorrelation method (e.g. adversarial networks)
- Application of similar decorrelation to BDT
- Application within a fully fledged analysis (including systematics etc.)

Conclusion & Outlook
- Introduced set of low level continuum suppression variables
- Prepared BDT and DNNs using introduced variables, expecting DNN to profit from those
- DNNs require decorrelation, which most likely limits their performance
- Fits on MC show similar accuracies for BDT/DNN but slighly better than BDT with common CS variables
→ Low level CS variables could reduce statistical errors but further investigation (e.g. systematics etc.) needed for final judgement
For the Future
- Study influence of single variables on sculpting (to possibly exclude them)
- Impact on performance with alternative decorrelation method (e.g. adversarial networks)
- Application of similar decorrelation to BDT
- Application within a fully fledged analysis (including systematics etc.)

1. In fact the sculpting also happens with only *engineered* variables. It's just that so far everyone always used BDTs which are not subject to that issue.

Backup

Backup

Data Samples

- Generic (run independent) MC ($q\bar{q}$ where $q = \mathrm{u,d,s,c}$ & $\mathrm{B\overline{B}}$): $1\,\mathrm{ab}^{-1}$
- Pure signal MC for signal channel and control channel: 4×10^6 and 2×10^6 events produced resulting in $1\,019\,638$ and $523\,183$ reconstructed events respectively
- Physics data: $361.65\,\mathrm{fb}^{-1}$
- Off-resonance generic MC ($q\bar{q}$ where $q = \mathrm{u,d,s,c}$): $169.328\,\mathrm{fb}^{-1}$
- Off-resonance data: $42.28\,\mathrm{fb}^{-1}$

MC Modeling

- Problems with the available samples ($\tau^- \ \tau^+$, momentum corrections) remain
- MC modeling overall not bad, considering the above

$\rightarrow$ Further investigation needed for final judgment

Data Samples
- Generic (run independent) MC ($q\bar{q}$ where $q = \mathrm{u,d,s,c}$ & $\mathrm{B\overline{B}}$): $1\,\mathrm{ab}^{-1}$
- Pure signal MC for signal channel and control channel: 4×10^6 and 2×10^6 events produced resulting in $1\,019\,638$ and $523\,183$ reconstructed events respectively
- Physics data: $361.65\,\mathrm{fb}^{-1}$
- Off-resonance generic MC ($q\bar{q}$ where $q = \mathrm{u,d,s,c}$): $169.328\,\mathrm{fb}^{-1}$
- Off-resonance data: $42.28\,\mathrm{fb}^{-1}$
MC Modeling
- Problems with the available samples ($\tau^- \ \tau^+$, momentum corrections) remain
- MC modeling overall not bad, considering the above
$\rightarrow$ Further investigation needed for final judgment

Continuum Suppression Variables

All Input Variables

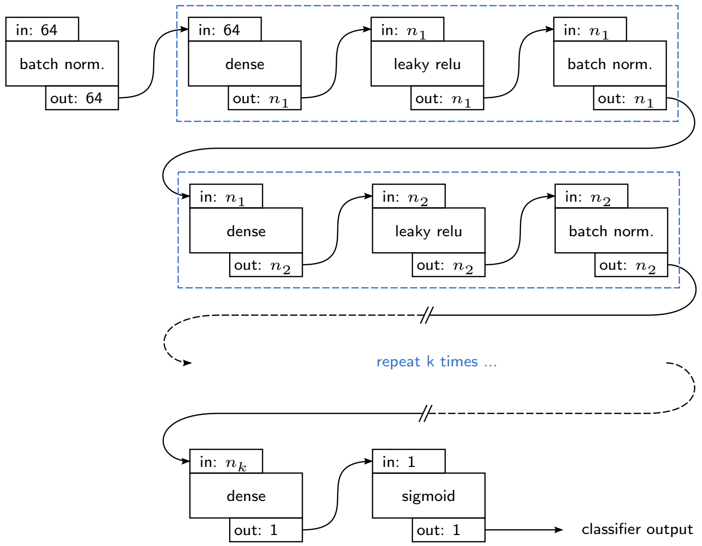
Δz	$R(\cos(\theta_p^{+s0}))$	$R(\phi_p^{0s0})$	$S(\cos(\theta_p^{-s0}))$	$S(\phi_d^{-r0})$	$S(p^{+s0})$
$\cos(\theta_{SR})$	$R(\cos(\theta_d^{-r0}))$	$R(\phi_p^{0s1})$	$S(\cos(\theta_p^{+r0}))$	$S(\phi_d^{-s0})$	$S(\phi_p^{0r0})$
$\cos(\theta_{Sz})$	$R(\cos(\theta_d^{-s0}))$	$R(\phi_p^{-r0})$	$S(\cos(\theta_p^{+s0}))$	$S(\phi_d^{+r0})$	$S(\phi_p^{0r1})$
M'_{bc}	$R(\cos(\theta_d^{+r0}))$	$R(\phi_p^{-s0})$	$S(\cos(\theta_d^{-r0}))$	$S(\phi_d^{+s0})$	$S(\phi_p^{0s0})$
$R(\cos(\theta_p^{0r0}))$	$R(\cos(\theta_d^{+s0}))$	$R(\phi_p^{+r0})$	$S(\cos(\theta_d^{-s0}))$	$S(p^{0r0})$	$S(\phi_p^{0s1})$
$R(\cos(\theta_p^{0r1}))$	$R(\phi_d^{-r0})$	$R(\phi_p^{+s0})$	$S(\cos(\theta_d^{+r0}))$	$S(p^{0r1})$	$S(\phi_p^{-r0})$
$R(\cos(\theta_p^{0s0}))$	$R(\phi_d^{-s0})$	$S(\cos(\theta_p^{0r0}))$	$S(\cos(\theta_d^{+s0}))$	$S(p^{0s0})$	$S(\phi_p^{-s0})$
$R(\cos(\theta_p^{0s1}))$	$R(\phi_d^{+r0})$	$S(\cos(\theta_p^{0r1}))$	$S(d^{-r0})$	$S(p^{0s1})$	$S(\phi_p^{+r0})$
$R(\cos(\theta_p^{-r0}))$	$R(\phi_d^{+s0})$	$S(\cos(\theta_p^{0s0}))$	$S(d^{-s0})$	$S(p^{-r0})$	$S(\phi_p^{+s0})$
$R(\cos(\theta_p^{-s0}))$	$R(\phi_p^{0r0})$	$S(\cos(\theta_p^{0s1}))$	$S(d^{+r0})$	$S(p^{-s0})$	
$R(\cos(\theta_p^{+r0}))$	$R(\phi_p^{0r1})$	$S(\cos(\theta_p^{-r0}))$	$S(d^{+s0})$	$S(p^{+r0})$	

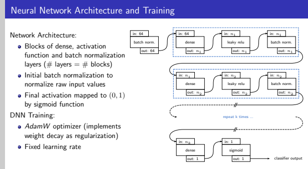
Network Architecture:

- Blocks of dense, activation function and batch normalization layers (# layers = # blocks)
- Initial batch normalization to normalize raw input values
- Final activation mapped to $(0, 1)$ by sigmoid function

DNN Training:

- *AdamW* optimizer (implements weight decay as regularization)
- Fixed learning rate

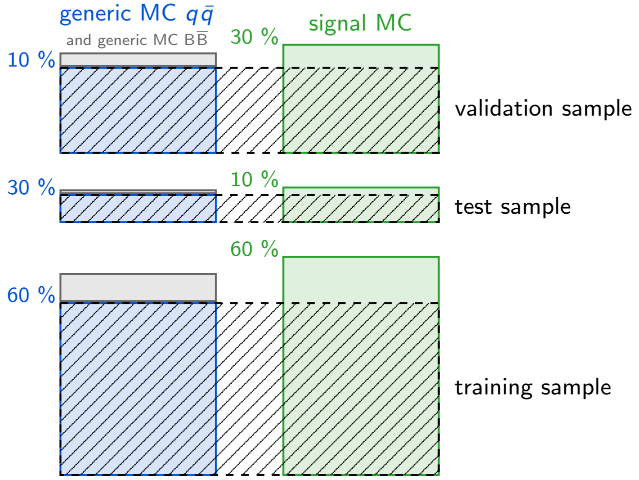


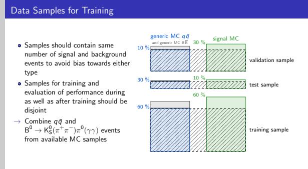


Data Samples for Training

- Samples should contain same number of signal and background events to avoid bias towards either type
- Samples for training and evaluation of performance during as well as after training should be disjoint

→ Combine $q\bar{q}$ and $\mathrm{B}^0 \to \mathrm{K}^0_\mathrm{S}(\pi^+\pi^-)\pi^0(\gamma\gamma)$ events from available MC samples

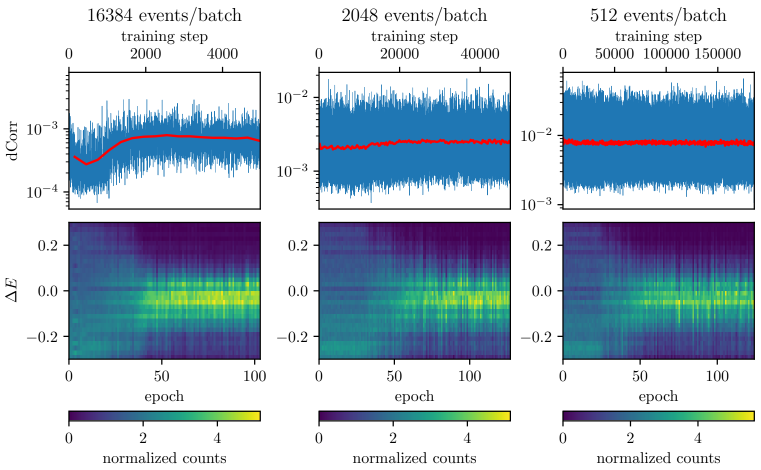


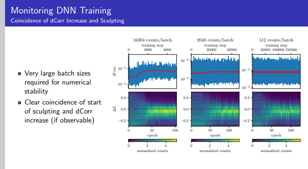


Monitoring DNN Training

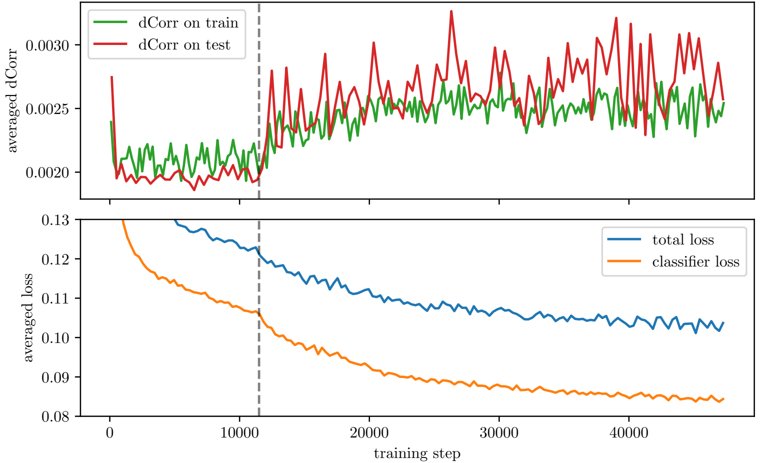
Coincidence of dCorr Increase and Sculpting

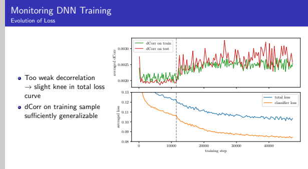
- Very large batch sizes required for numerical stability
- Clear coincidence of start of sculpting and dCorr increase (if observable)





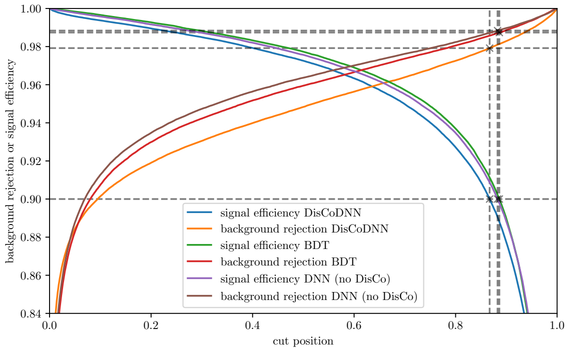
- Too weak decorrelation
  → slight knee in total loss curve
- dCorr on training sample sufficiently generalizable

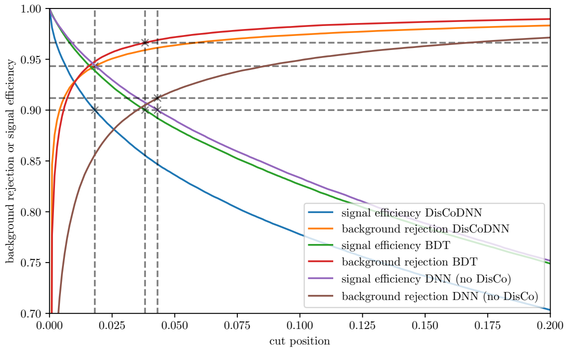


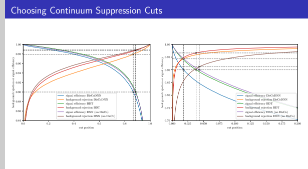


1. Talk about intuition of *barrier* in parameter space. DisCo appear to introduce barrier but never really plane the global (correlated) minimum.

Choosing Continuum Suppression Cuts







	signal	$q\bar{q}$	$\mathrm{B\overline{B}}$
true yield DisCoDNN	318	3313	71
true yield BDT	321	2134	75
yield DisCoDNN	310.6 ± 28.3	3343 ± 39	49.30 ± 31.28
yield BDT	337.5 ± 26.1	2149 ± 35	43.52 ± 27.83
rel. fit error DisCoDNN in %	8.902	1.178	44.06
rel. fit error BDT in %	8.144	1.626	37.1
rel. true error DisCoDNN in %	2.335 ± 8.902	0.897 ± 1.178	30.57 ± 44.06
rel. true error BDT in %	5.133 ± 8.144	0.710 ± 1.626	41.97 ± 37.10
pull DisCoDNN in σ	-0.2623	0.7619	-0.6937
pull BDT in σ	0.6302	0.4367	-1.131

Fits on MC
Fit Results Table

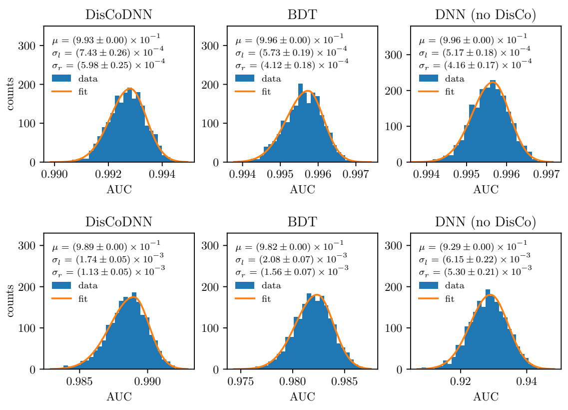
Classifier Performance Stability & Input Variable Correlations

Bootstrapping

- Models fluctuations of occurrences of event types, not numerical fluctuations
- All classifiers remain reasonably stable

Uncorrelated Toys

- Do not model correlations, as nearly impossible
- Classifiers that do not significantly sculpt ΔE barely utilize correlations between input variables



2023-12-19

CS with NNs for Belle II

└Classifier Performance Stability & Input Variable Correlations