SCHOOL OF DATA ANALYSIS

NATIONAL RESEARCH UNIVERSITY

Topological Trigger Optimization

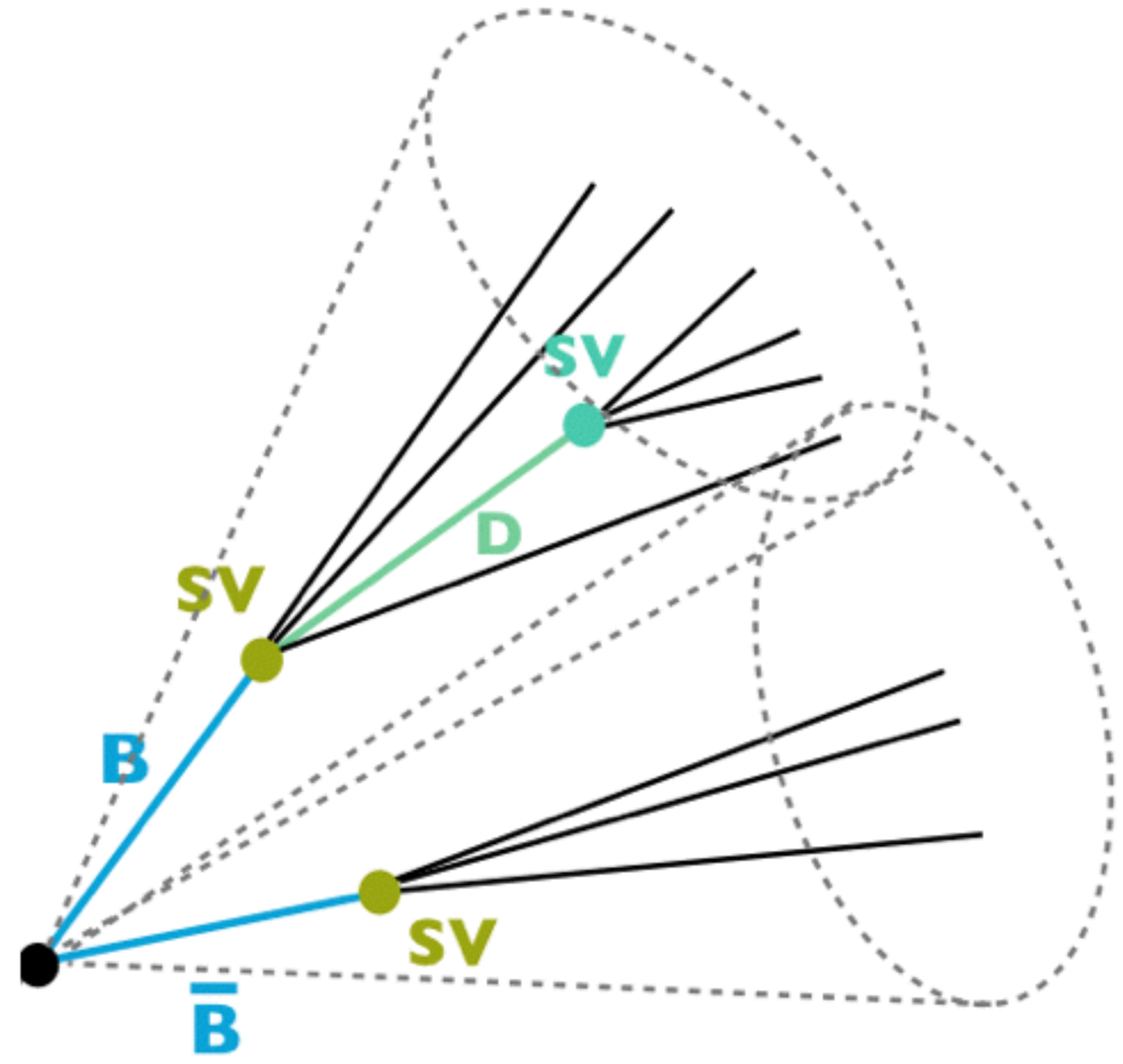
Tatiana Likhomanenko

YSDA, NIRC «Kurchatov Institute», HSE

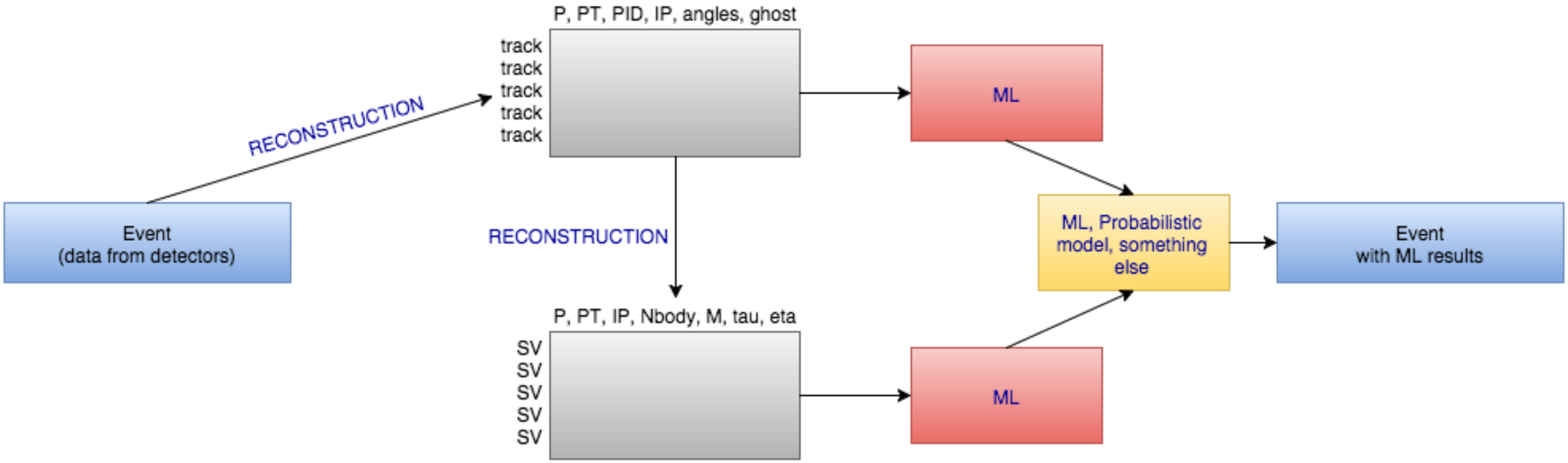
Data structure

Event

- Sample: one proton-proton bunches collision
- Event consists of:
  - tracks (track description)
  - secondary vertices (SV description)
- Questions:
  - How to describe event in ML terms?
  - How to train model on such events?



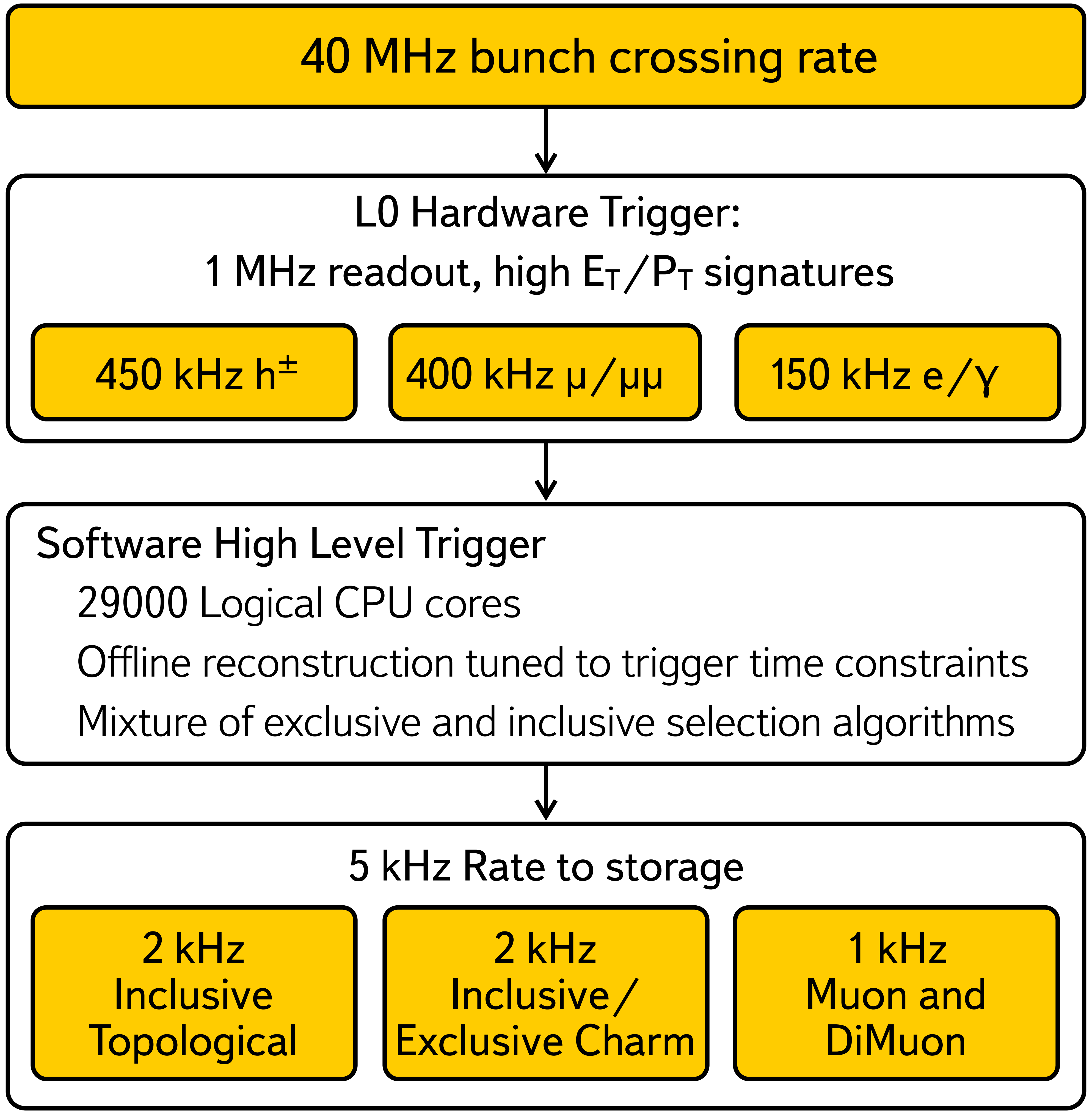
ML in event processing



Topological trigger

LHCb trigger system

- Select events to store them for offline processing
- Should efficiently select interesting events
- An event is interesting if it contains at least one interesting SV
- Output rate for trigger system is limited

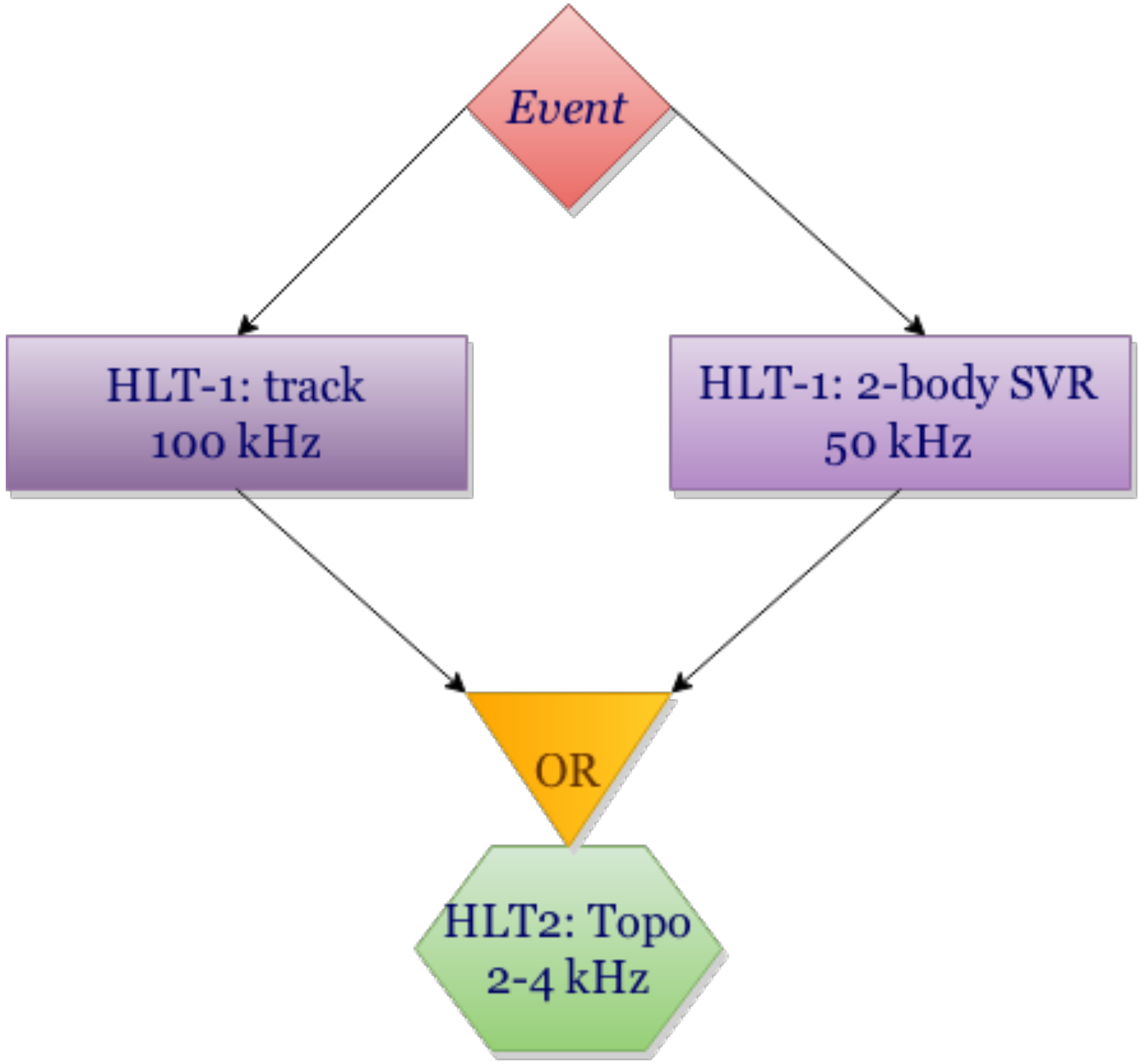


LHCb topological trigger

- Generic trigger for decays of beauty and charm hadrons
- It designed to be inclusive trigger line to efficiently select any B decay with at least 2 charged daughters
- Look for 2, 3, 4 track combinations in a wide mass range
- Designed to efficiently select decays with missing particles
- Use fast-track fit to improve signal efficiency and minbias rejection

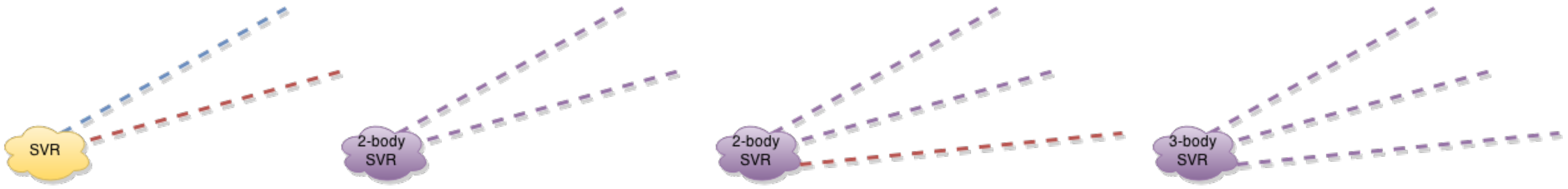
Run-II LHCb topological trigger

- HLT-1 track is looking for either one super high PT or high displacement track
- HLT-1 2-body SV classifier is looking for two tracks making a vertex
- HLT-2 improved topo classifier uses full reconstructed event to look for 2, 3, 4 and more tracks making a vertex



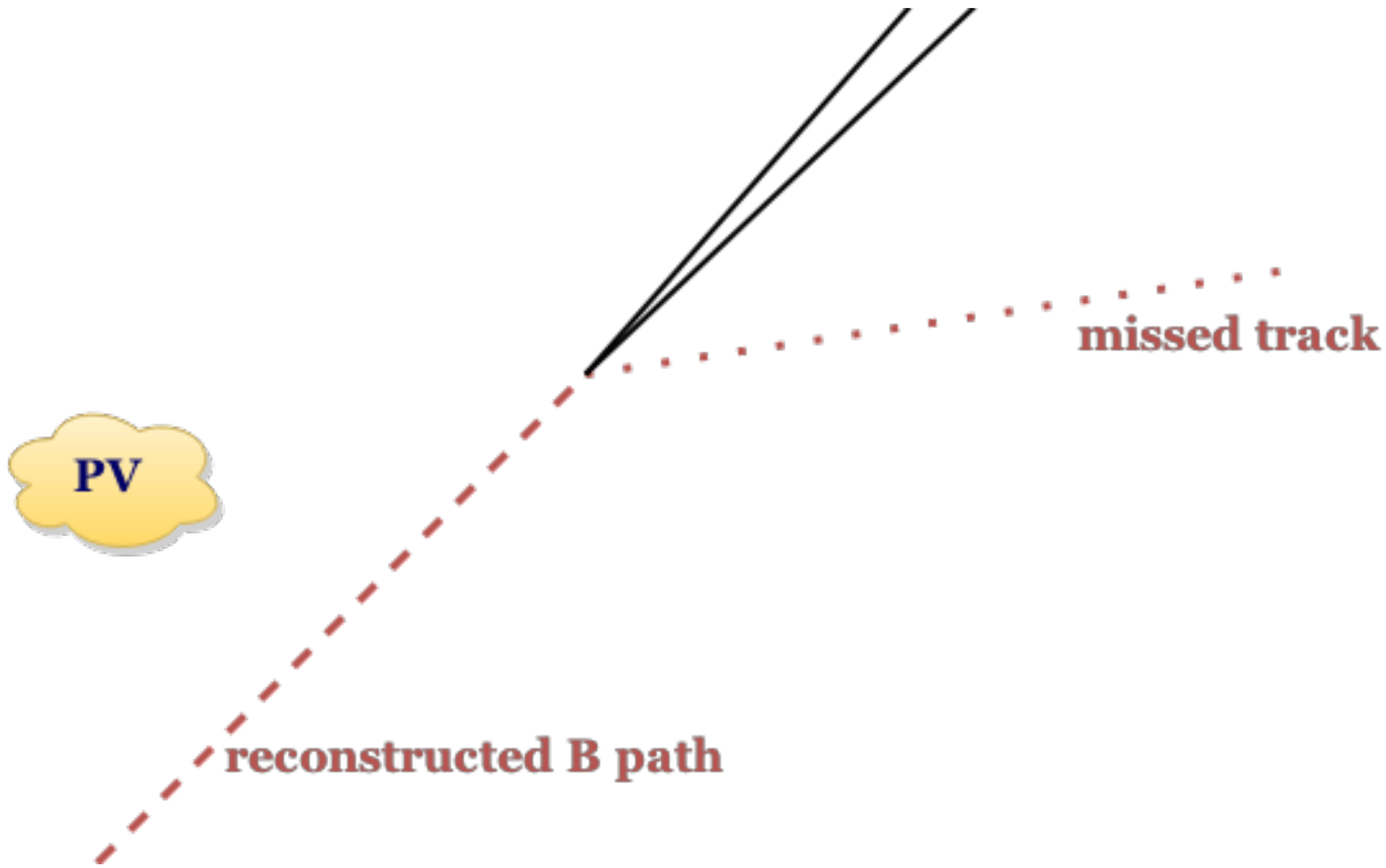
N body tracks

- Two, three or four tracks are combined to form a SVR
- Each secondary vertex in Monte Carlo data is preselected in such way, that all tracks must be matched to particles from the signal decay (true match preselection)

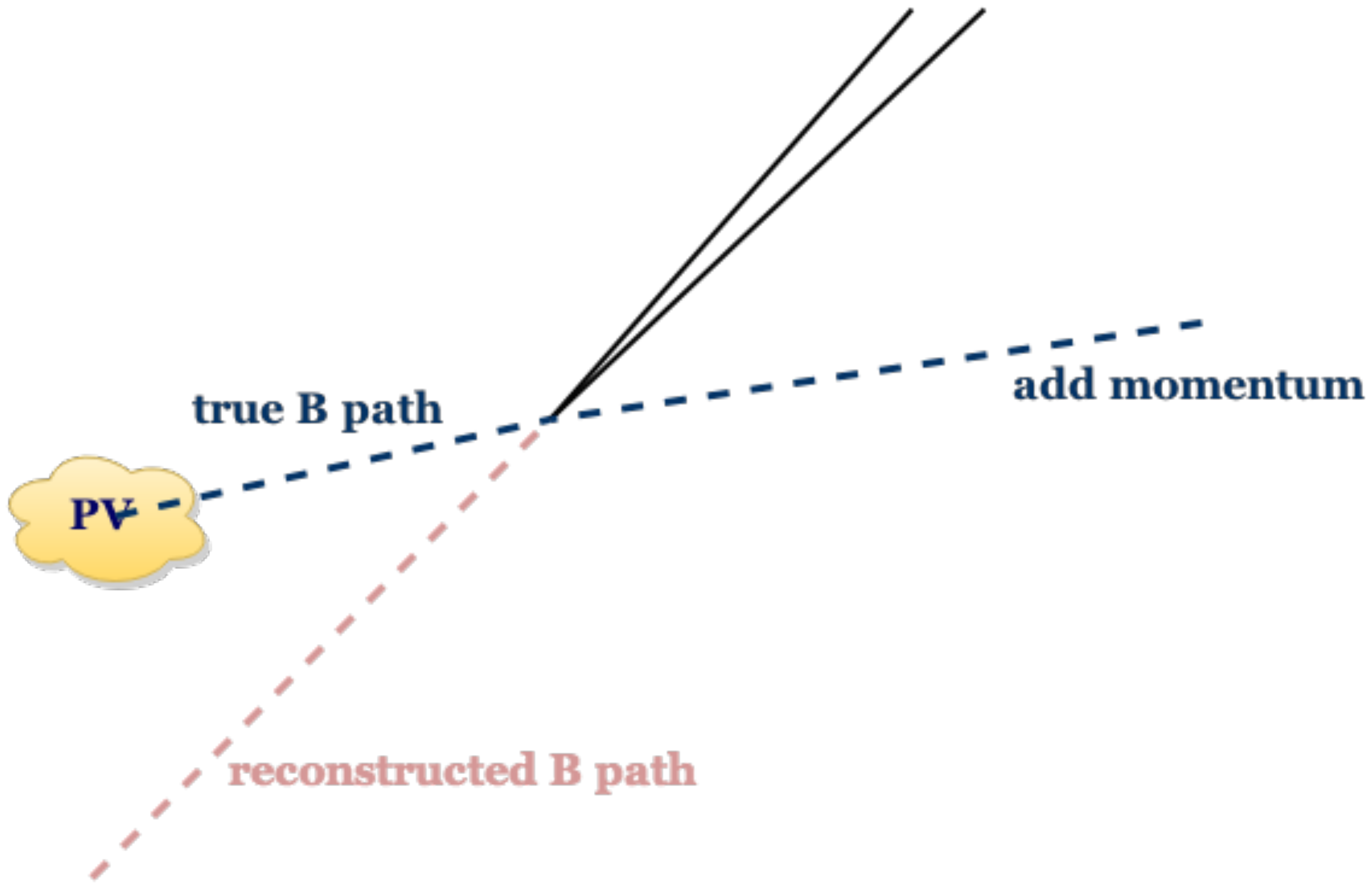


Omission of daughters

- The trigger is designed to allow for the omission of one or more daughters when forming the trigger candidate



Omission of daughters

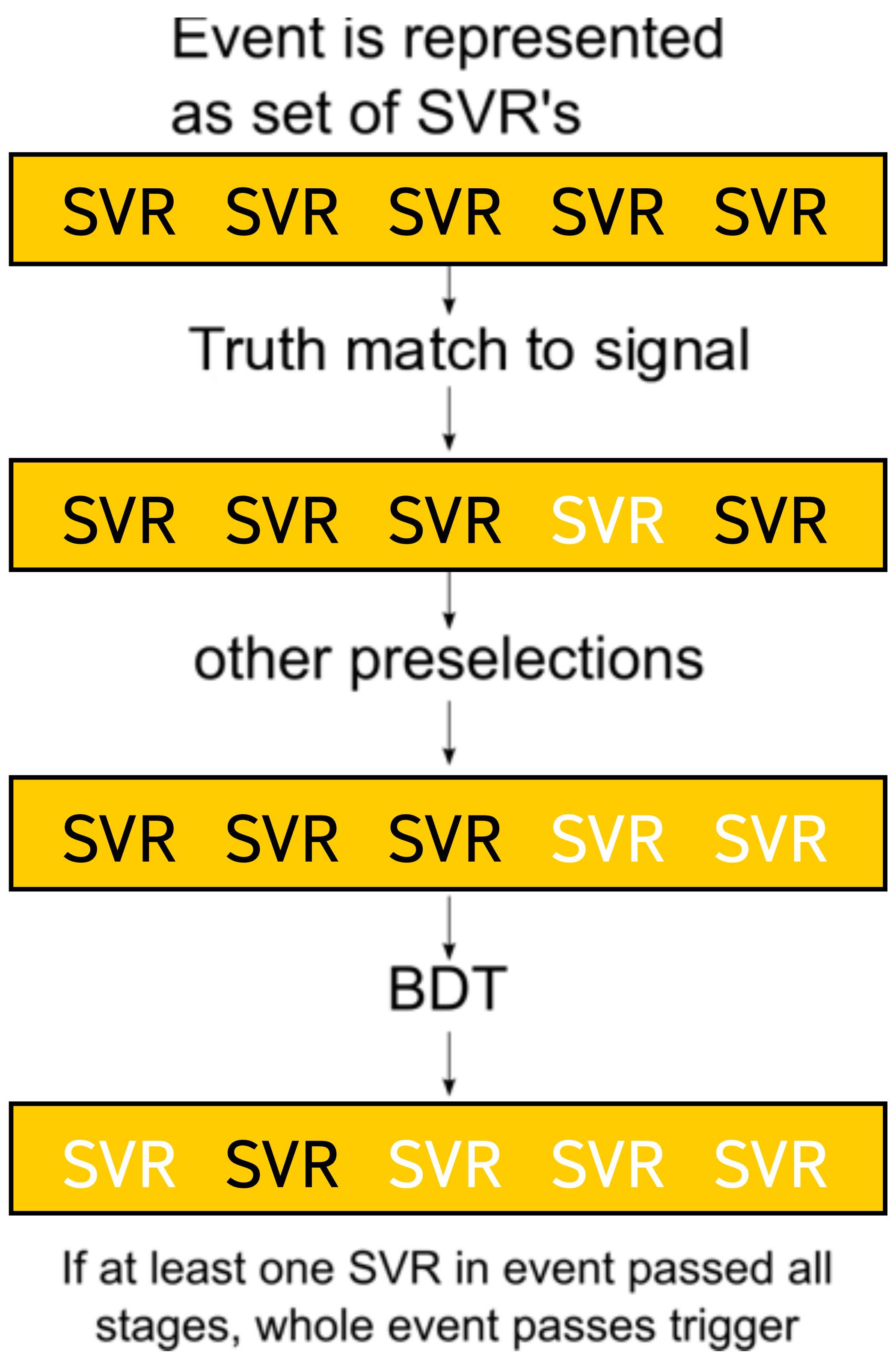


$$m_{corrected} = \sqrt{m^2 + \left|p'_{Tmissing}\right|^2} + \left|p'_{Tmissing}\right|$$

Machine learning problem

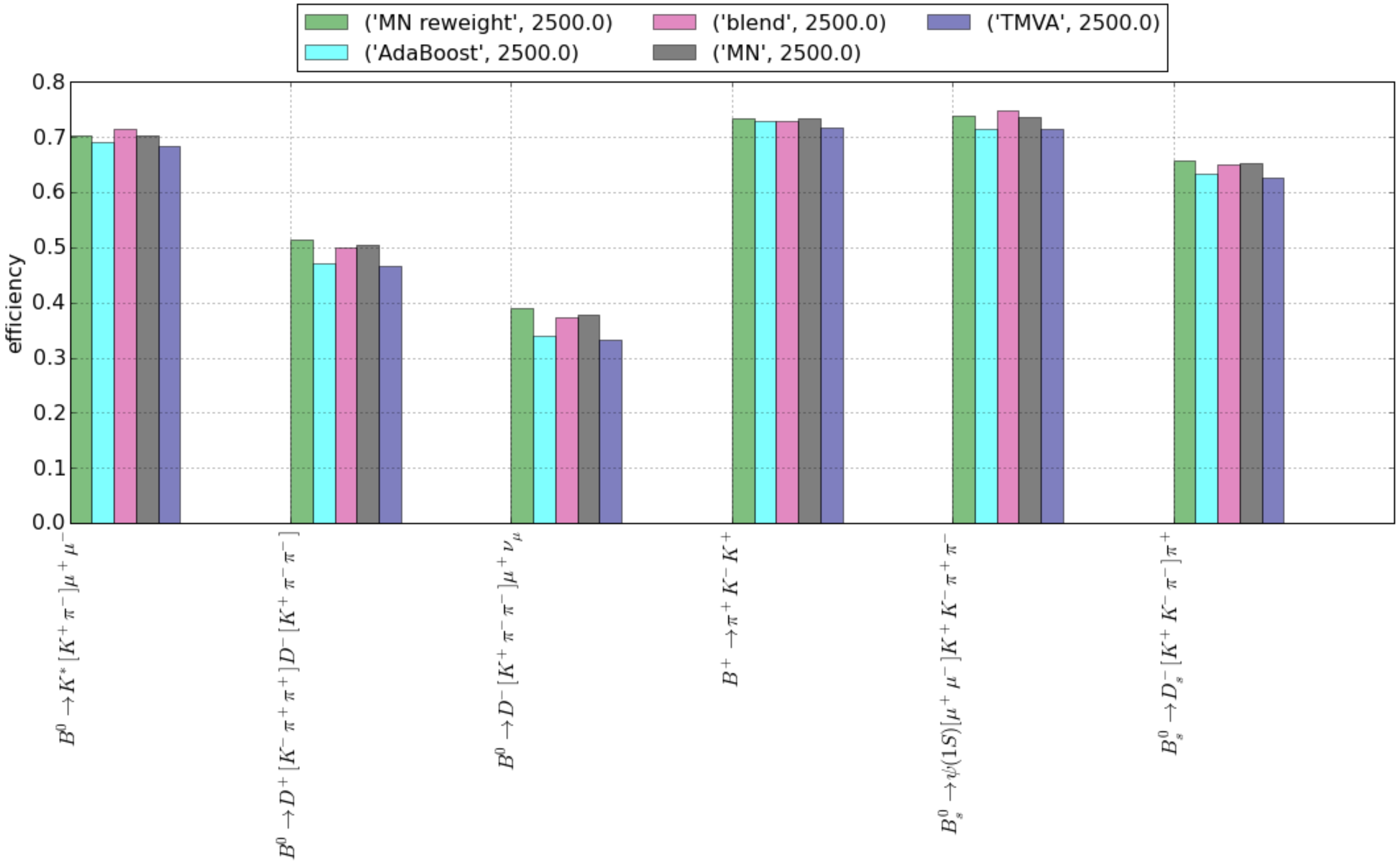
Data

- Monte Carlo samples (used as signal-like) are simulated 13-TeV B decays of various topologies
- Generic Pythia 13-TeV proton-proton collisions are used as background-like sample
- Training data are set of SVs for all events
- Most events have many secondary vertices (not all events have them)
- Goal is to improve efficiency for each type of signal events along fixed efficiency for background



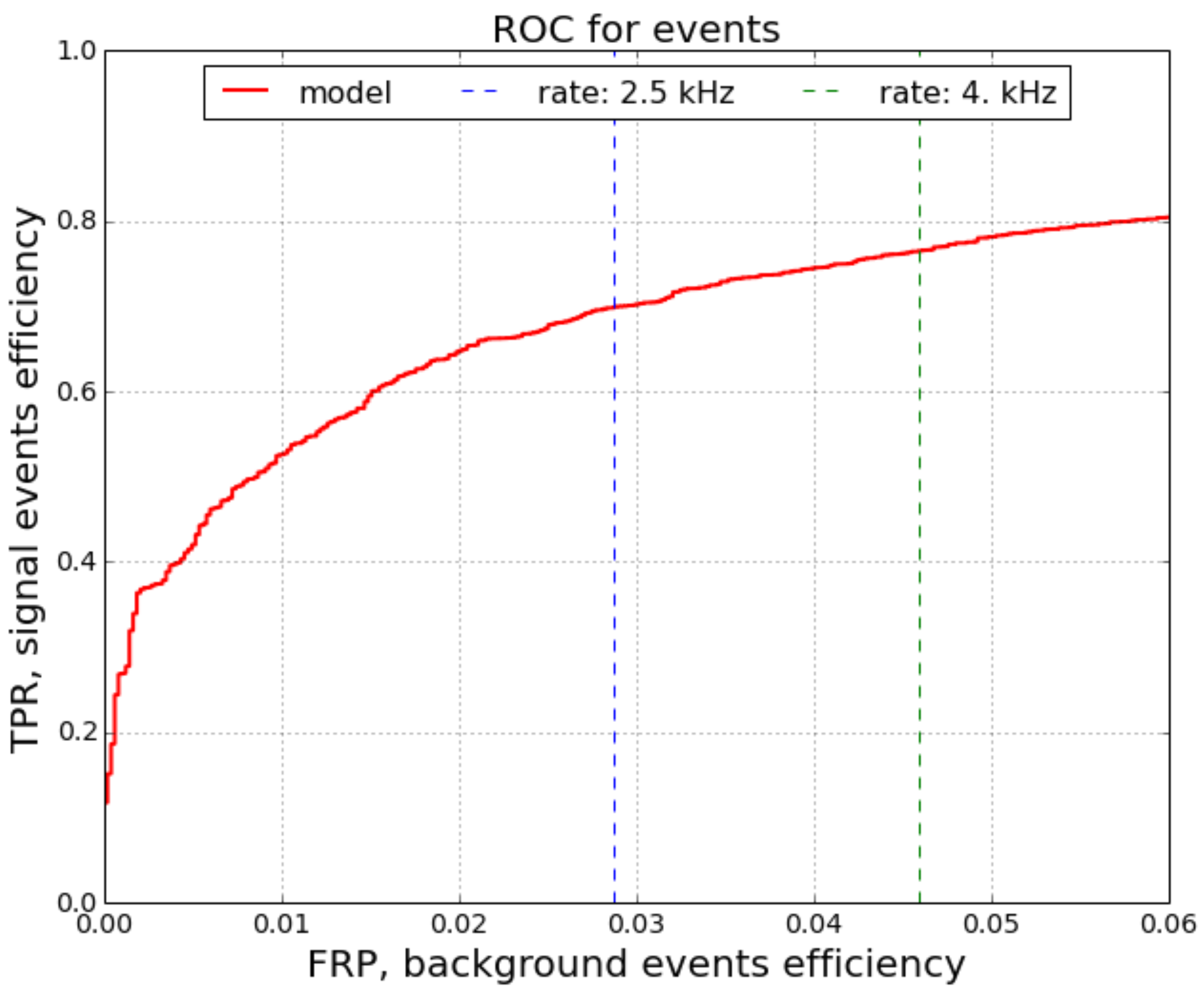
Event representation

How to measure quality?



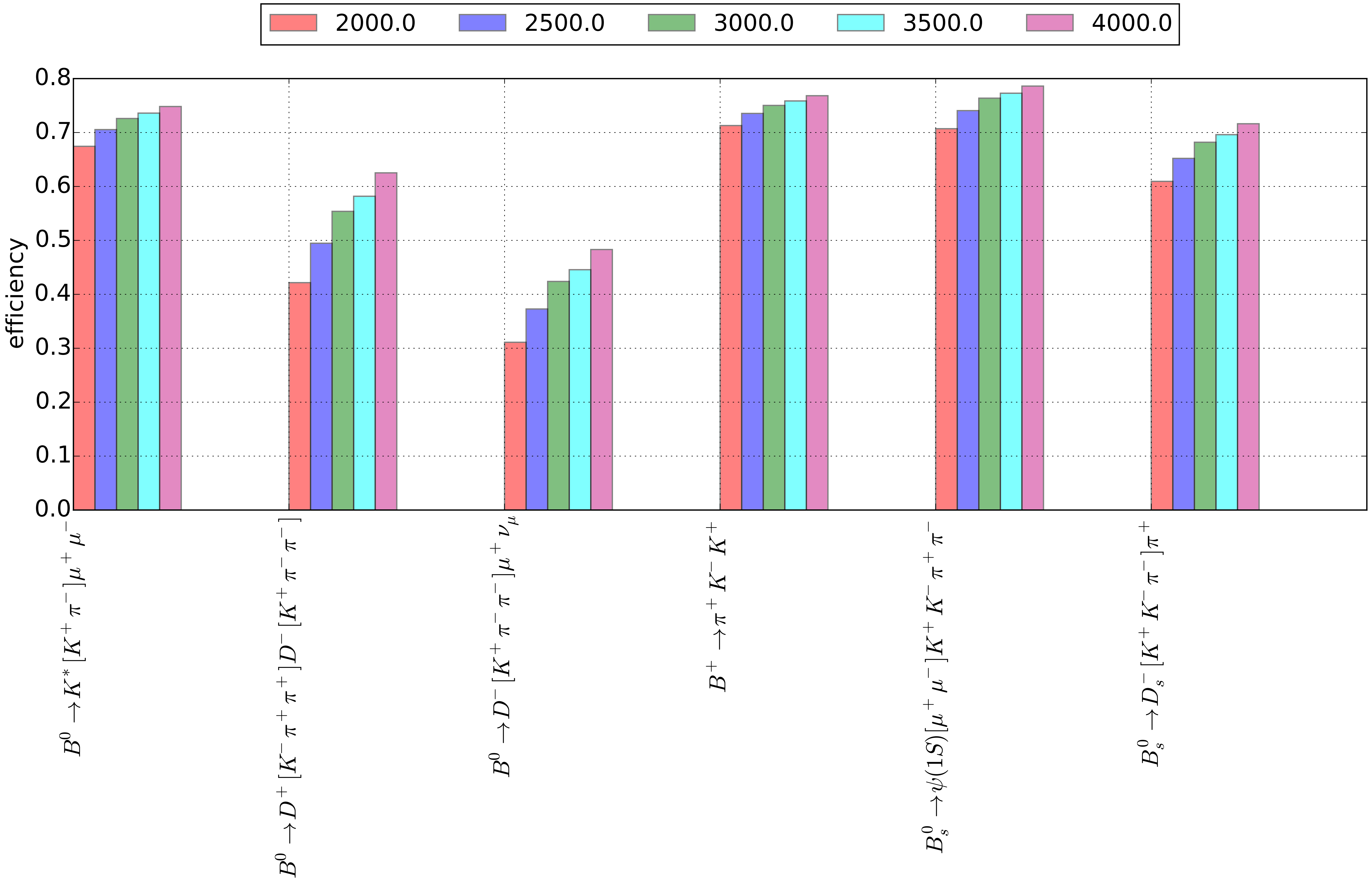
ROC curve, computed for events

- Output rate = false positive rate (FPR) for events
- Optimize true positive rate (TPR) for fixed FPR for events
- Weight signal events in such way that channels have the same amount of events.
- Optimize ROC curve in a small FPR region



ROC curve interpretation

Dependence on output rate



Hierarchical training

- Train separate models for:
  - each channel
  - each n-body type: 2, 3, 4
- Use them as additional features later
- Use folding scheme to train additional features or to apply additional classifier selections

Folding scheme

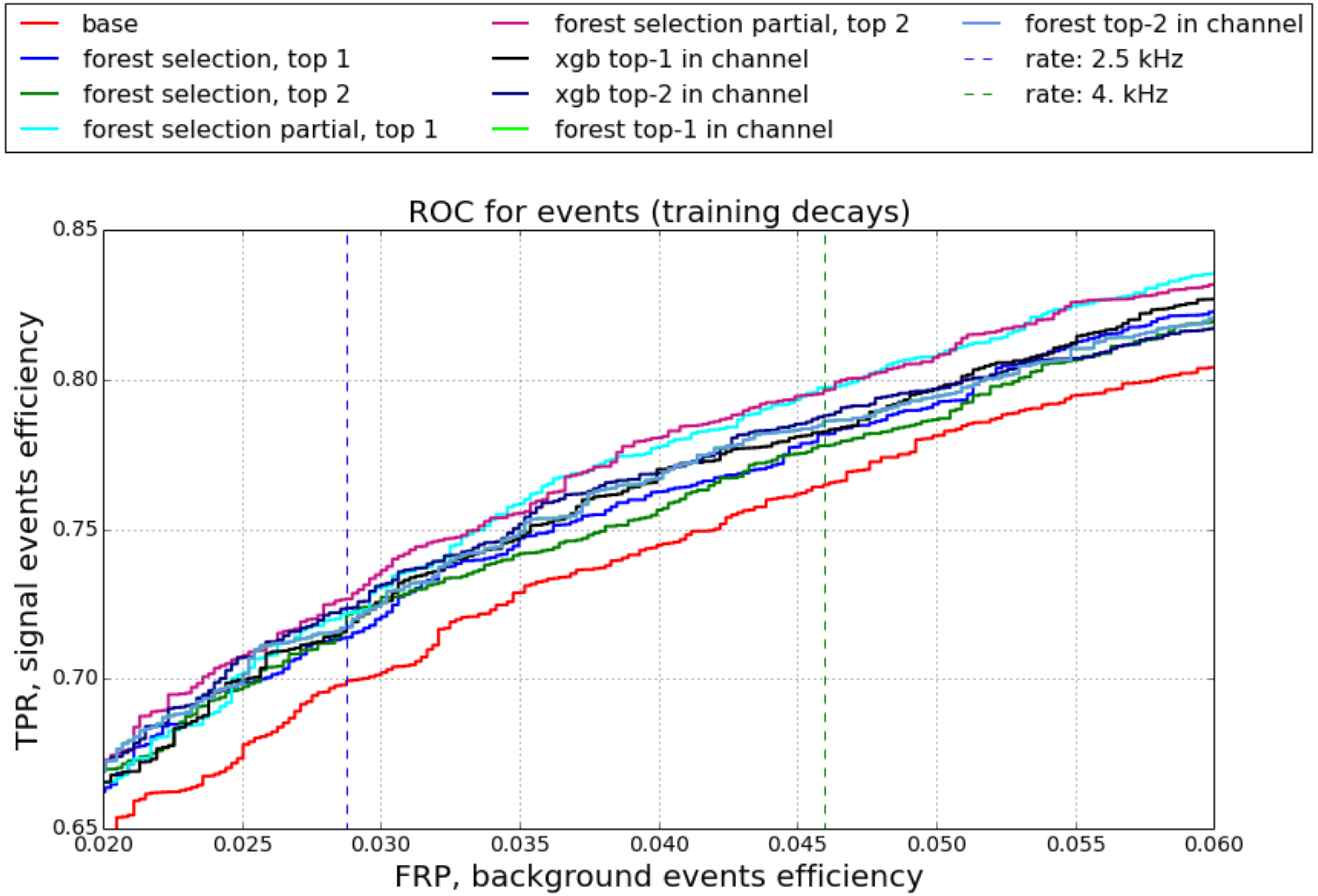
Random forest trick

Simulated signal event contains at least one interesting SV, but not each SV should be interesting

Random forest for SVs selection

- Train random forest (RF) on SVs using folding scheme
  - RF is stable to noise in data
  - RF doesn't penalize in case of misclassification (can find noisy samples)
- Select top-1, top-2 SVs by RF predictions for each signal event
- Train classifier on selected SVs
- Try another algorithm instead of RF, maybe it will work!

Random forest for SVs selection



Random forest for SVs selection: modifications

- Select top-1, top-2 SVs only for big channels, for small channels take all
- Train for each channel individual RF to select top-1, top-2 in the channel

Real-time

Online processing

There are two possibilities to speed up prediction operation:

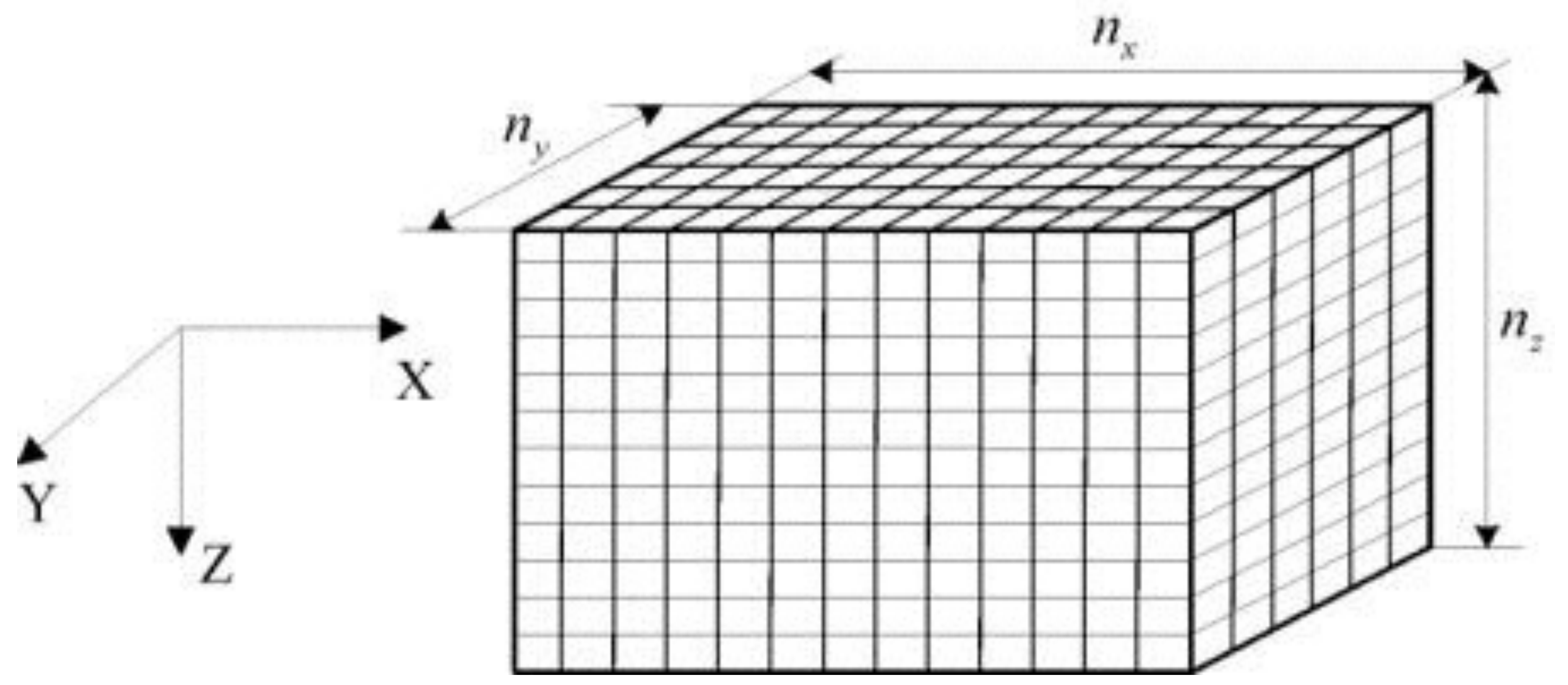
- Bonsai boosted decision tree format (BBDT)
- Post-pruning

Used algorithm (MatrixNet)

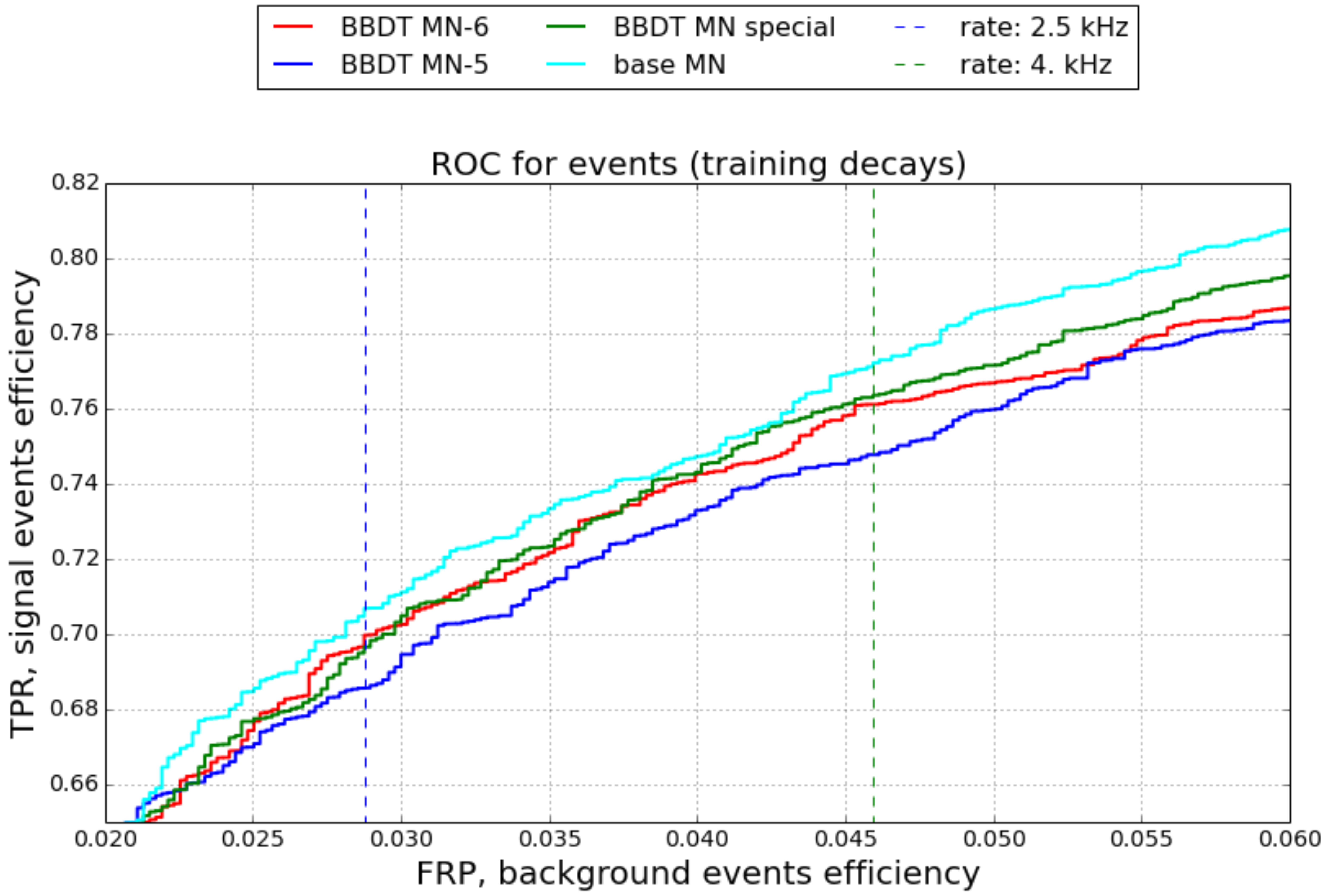
- Gradient Boosting over oblivious Decision Trees
- Feature binarization
- Classification, Regression, Ranking
- Yandex search engine exploits MatrixNet

BBDT

- Features hashing using bins before training
- Converting decision trees to
  n-dimensional table (lookup table)
- Table size is limited in RAM (1Gb), thus count of bins for each features should be small (5 bins for each of 12 features)
- Discretization reduces the quality
- Prediction operation takes one reading from the table



BBDT, results



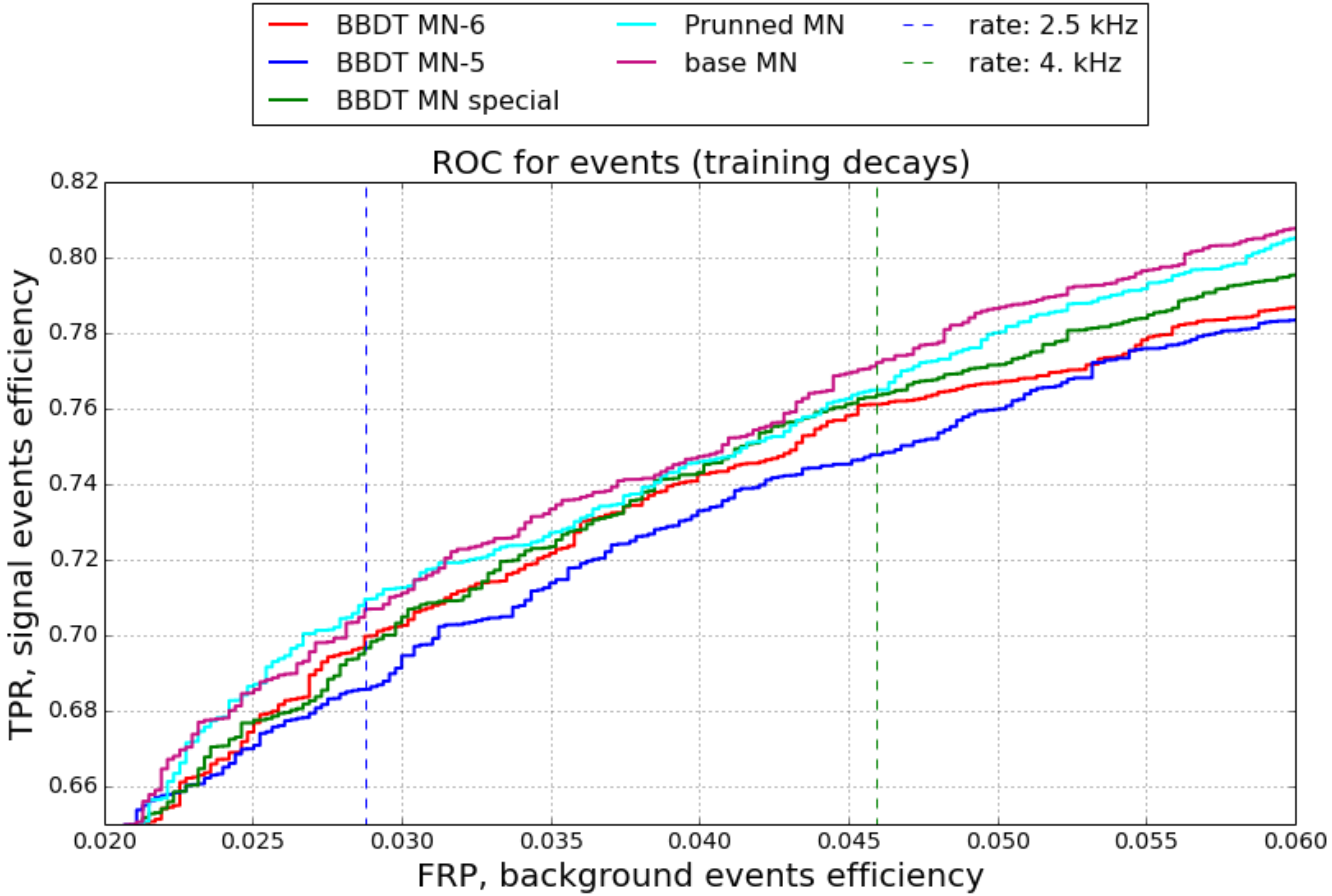
Post-pruning

- Train MatrixNet (MN) with several thousands trees
- Reduce this amount of trees to a hundred
- Greedily choose trees in a sequence from the initial ensemble to minimize a modified loss function:

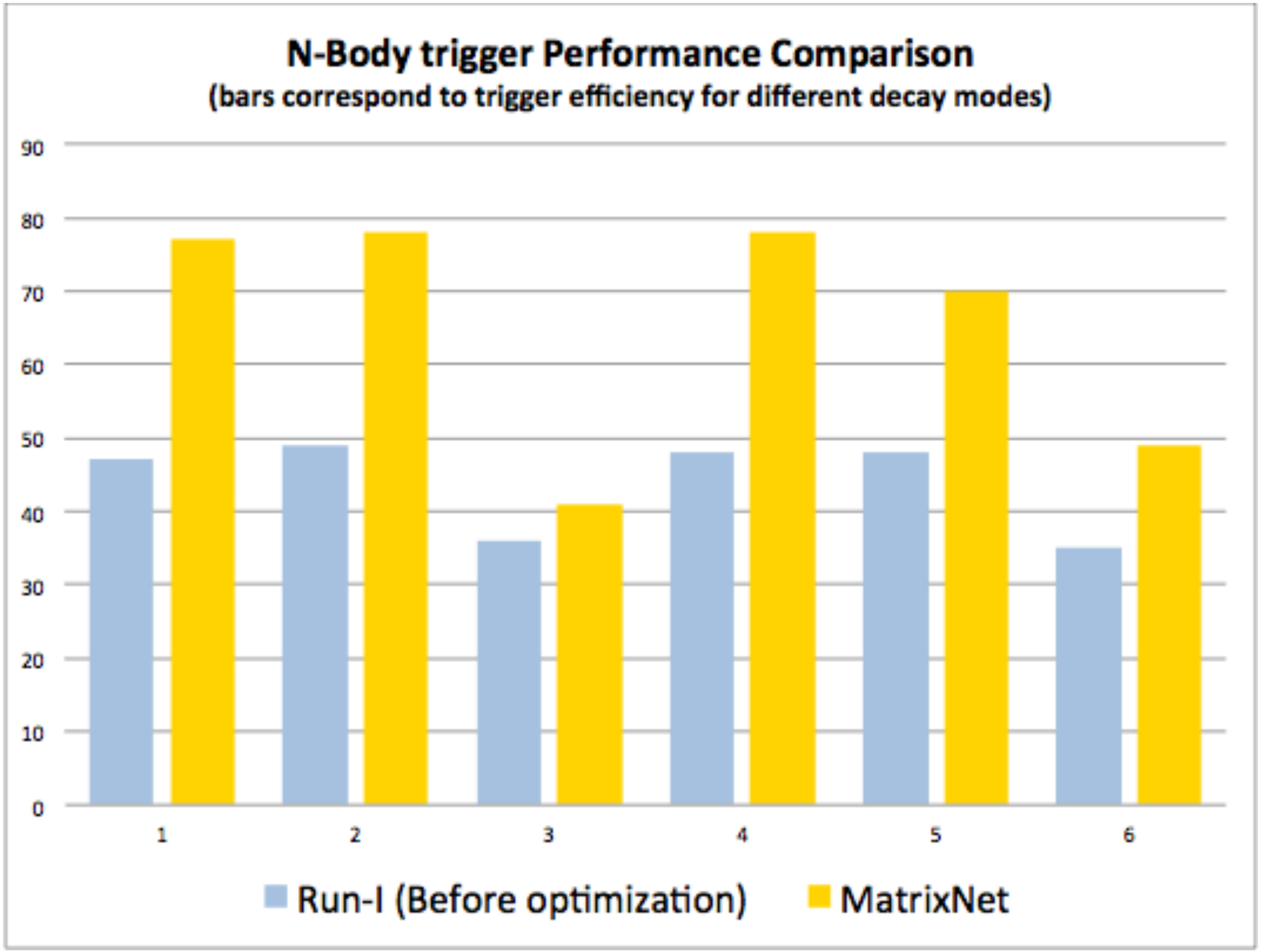
$$\sum_{signal} \log\left(1 + e^{-F(x)}\right) + \sum_{background} e^{F(x)}$$

- At the same time change values in leaves (tree structure is preserved)

Post-pruning, results



Topological trigger results (without RF trick)



https://github.com/yandexdataschool/LHCb-topo-trigger

References

- https://github.com/yandexdataschool/LHCb-topo-trigger
- https://cdsweb.cern.ch/record/1384380/files/LHCb-PUB-2011-016.pdf
- http://arxiv.org/abs/1510.00572

Thanks for attention

Contacts

Likhomanenko Tatiana
researcher-developer

antares@yandex-team.ru, tatiana.likhomanenko@cern.ch