General recipe to form input space for deep learning analysis of the collider hard scattering processes

SINP MSU
L. Dudko, M. Perfilov, P. Volkov, G. Vorotnikov

- Distinguishing of a scattering process is one of the main task in collider experiments
- 2→n particles hard process has (3n-4) independent variables, but the choice of observables for deep learning analysis is still an open question.
- Is it possible to formulate a general recipe to form optimal input space for deep learning analysis of the collider hard processes? **[Int.J.Mod.Phys.A 35 (2020) 21, 2050119, hep-ph:2002.09350]**

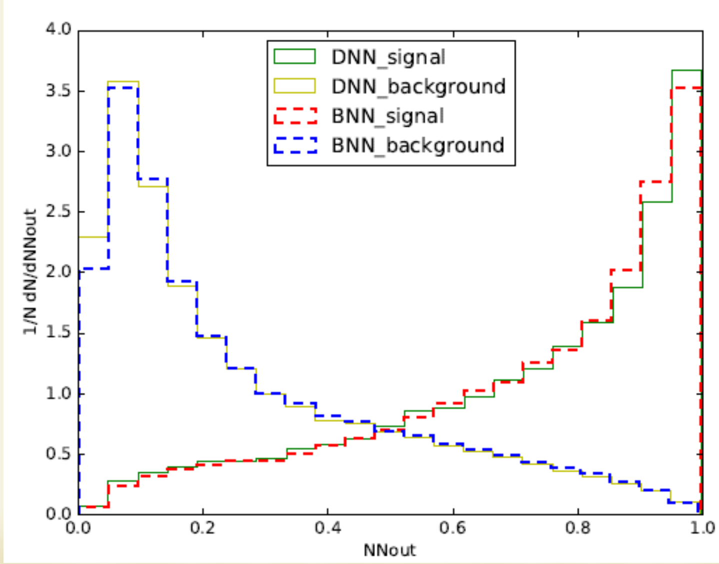
Method of "optimal observables" to find high level observables

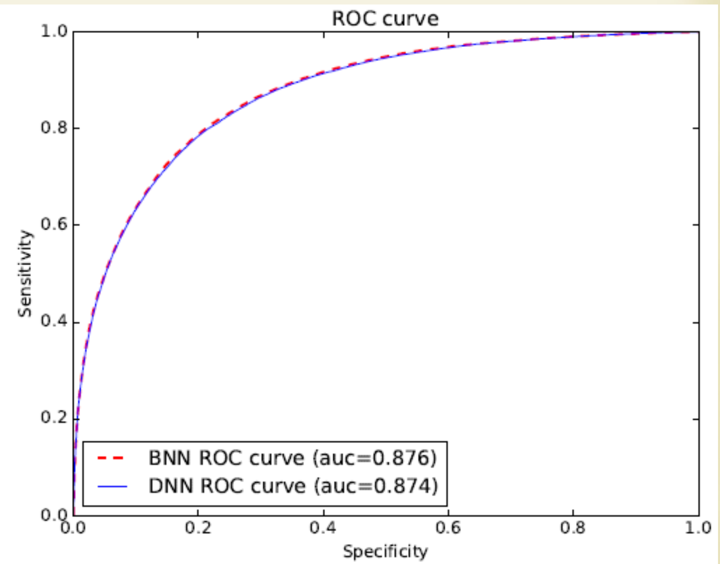
- **Developed for single top analysis in D0 (1998). Provides general recipe how to choose most sensitive high level variables to separate signal and background by NN**
  - It is based on the analysis of Feynman diagrams (FD) contributing to signal and background processes
  - Distinguish **three classes** of sensitive variables for the signal and each of kinematically different backgrounds: **Singular** variables (denominators of FD: s- and t-channel singularities), **Angular** variables (numerators of FD) and **Threshold** variables (Energy thresholds of the processes, H_T, ...)
  - Set of variables can be extended with other type of information, like detector relative variables (jet width, b-tagging discriminant, ...)

  Described in different examples for the single top and Higgs searches
  - D0-Note-3612 (1999) "Singularities of Feynman Diagrams and Optimal Kinematic Variables for Neural Networks."
  - E.Boos, V.Bunichev, L.Dudko, A.Markina, M.Perfilov **Phys.Atom.Nucl. 71 (2008) 388-393**
  - E.Boos, L.Dudko, T.Ohl Eur.Phys.J. C11 (1999) 473-484
  - E.Boos, L.Dudko Nucl.Instrum.Meth. A502 (2003) 486-488
- **Applied in different experimental analysis in D0 and CMS**
  - Phys.Lett. B517 (2001) 282-294 and some other single top D0 publications
  - JHEP02(2017)028 (CMS-TOP-14-007)

Move to Deep Learning NNs

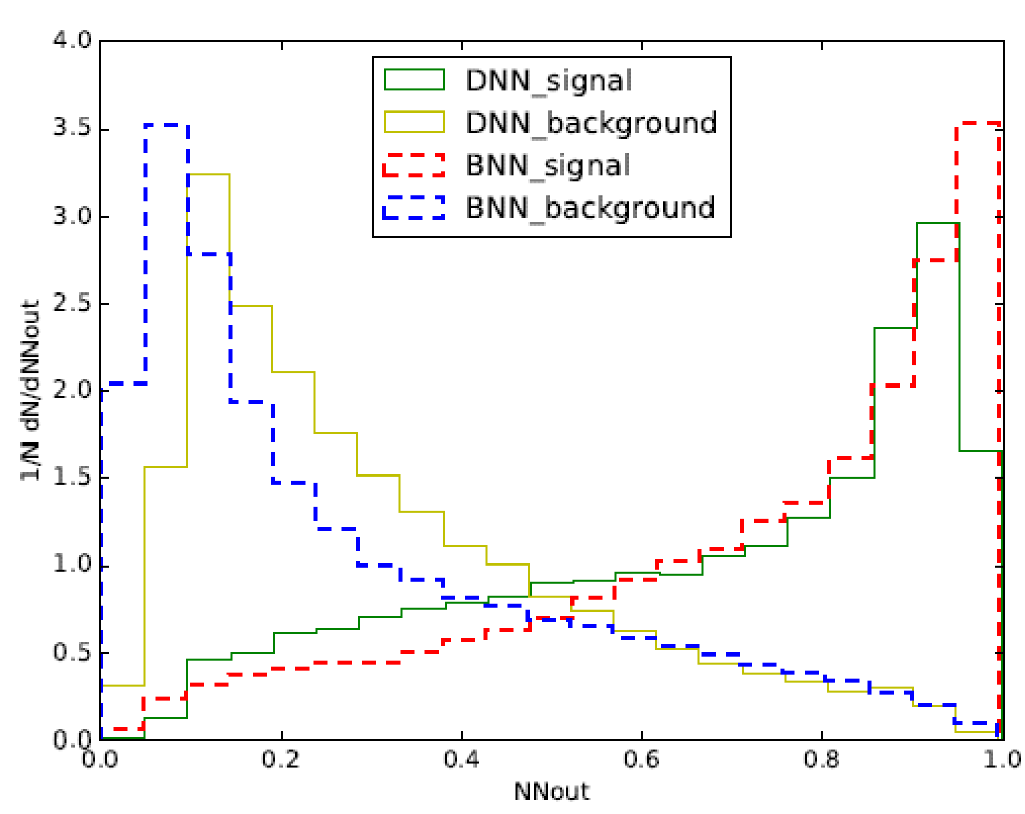
- Deep learning (DNN) makes possible to move from sophisticated input set of observables to some general raw level observables, and leave sensitive features extraction to DNN
- Differential cross sections $d\sigma \sim M^2(p_i \cdot p_f, s, t, u)$ is a function of 4-momenta. One needs to separate several functions of 4-momenta.
- Kolmogorov-Smirnov representation theorem can be used as a proof that it is possible in principle. NN is able to reproduce all continuous functions.
- As a benchmark we use distinguishing of single top quark production from pair top quark production. Not a trivial example, but very well investigated already: [**JHEP02(2017)028**] take the highly optimized set of optimal variables to compare efficiency
- First compare the methods for benchmark: Bayesian 1-layer NN (BNN, FBM package), as it is taken in CMS analysis, and DNN (Tensorflow) with the same set of high level variables:

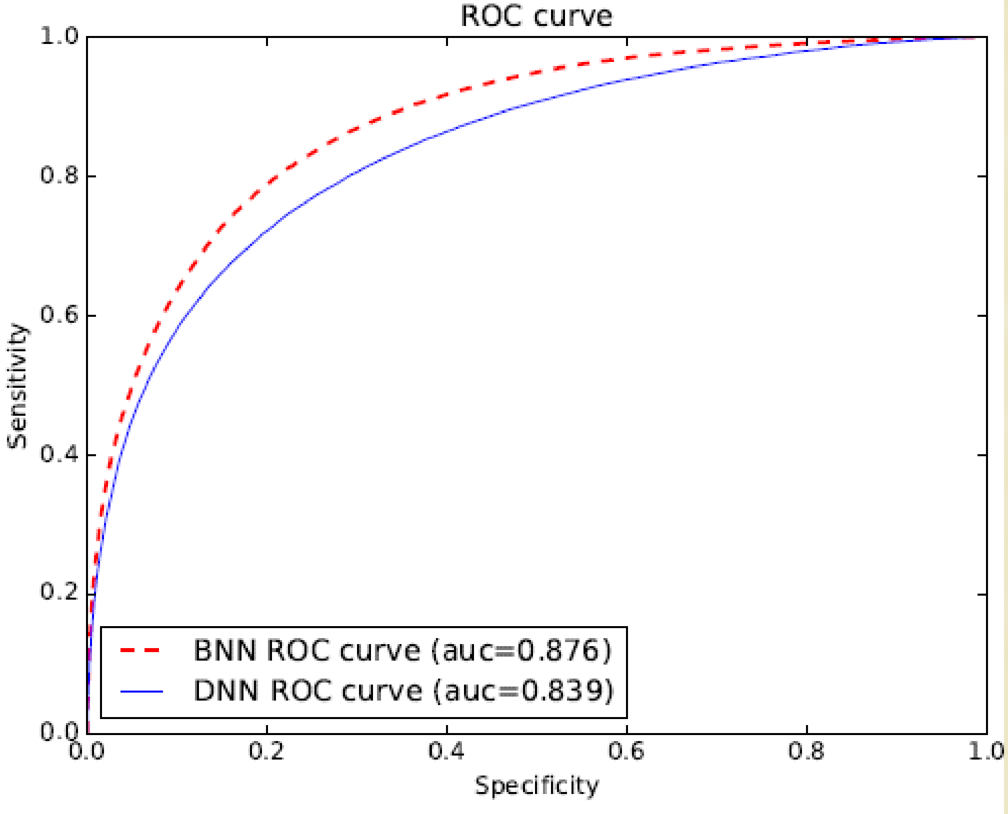




Check of the most simple set with Deep Learning NNs

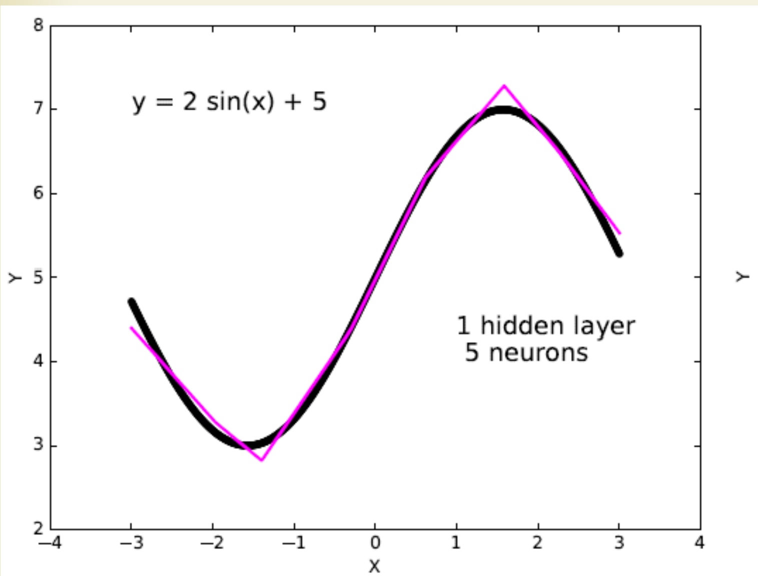
- For the first step one can take 4-momenta of the final particles as an input set for DNN
- Comparison of DNN (3 hidden layers) with benchmark BNN (trained with highly optimized set of high level variables) demonstrates not an optimal efficiency

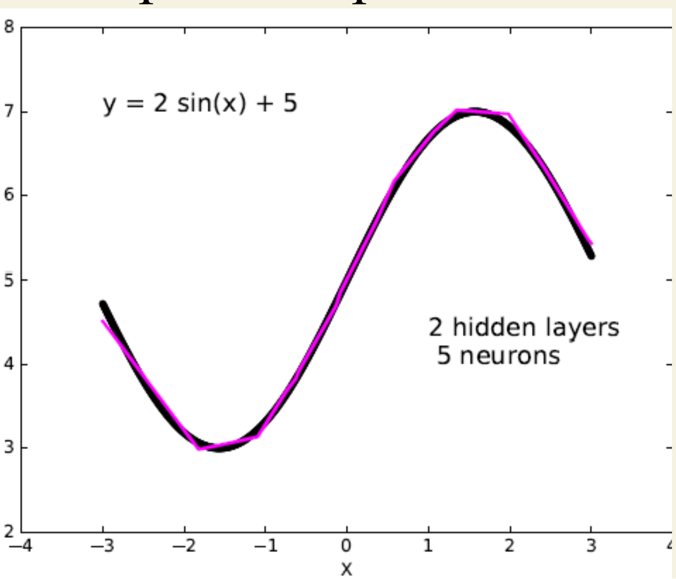


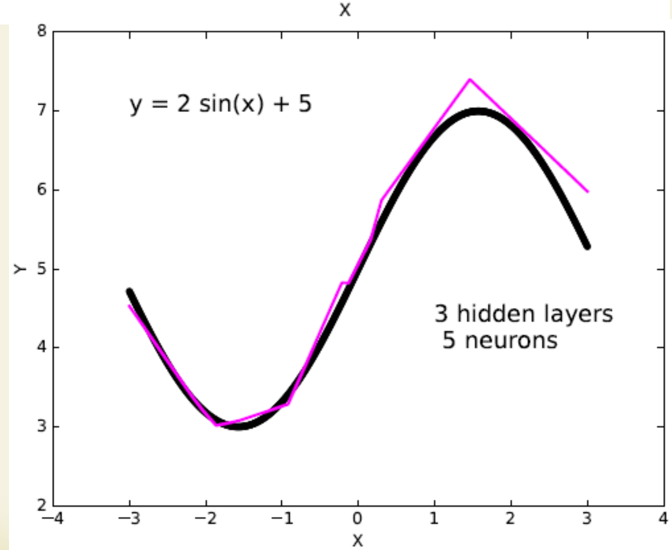


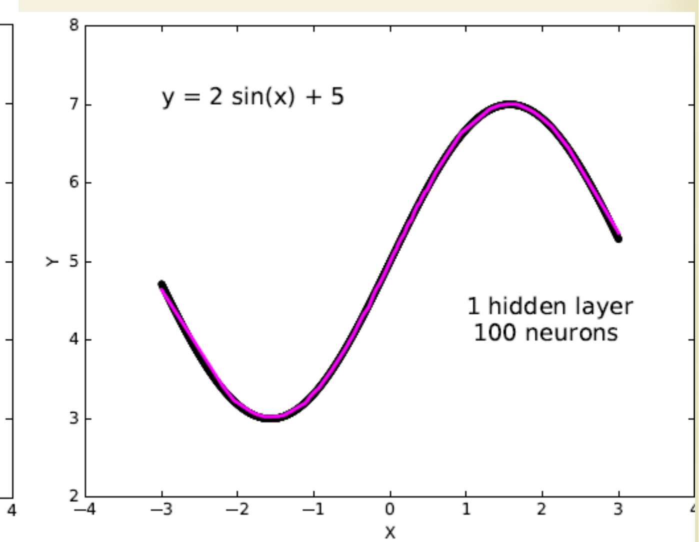
Why the 4-momenta is not an optimal general set? Linearity.

- The only non linear part of feed-forward NN is the activation function. M^2 is a function of scalar products of 4-momenta and quadratic forms (s,t,u), not 4-momenta. Therefore, it is needed to decrease the order of non-linearity or increase the number of hidden nodes. Simple example with ReLU to reproduce sin()







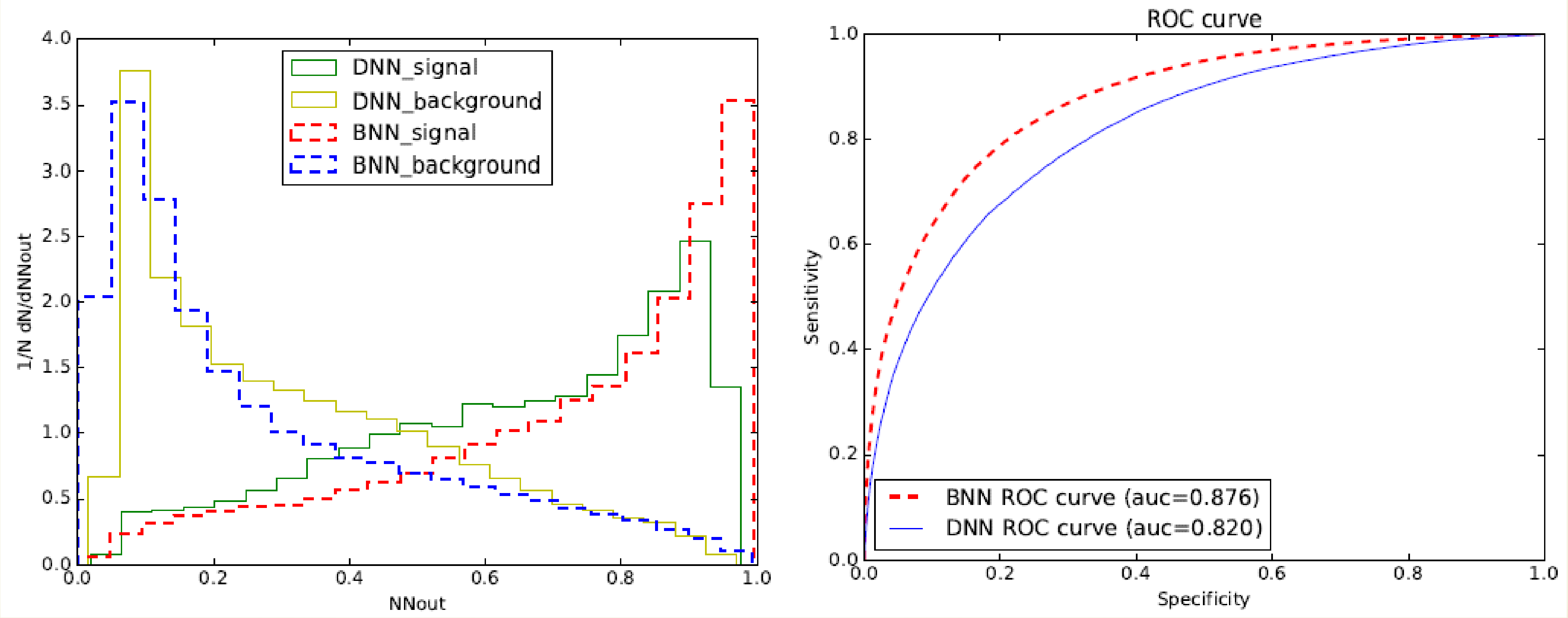


Check of quadratic forms of 4-momenta as an input for DNN

- An example of simple matrix element (s-channel ud→tb single top production) depends on scalar products of 4-momenta and Mandelstam variables (s,t,u in general case)

$$|M|^2 = V_{tb}^2 V_{ud}^2 (g_W)^4 \frac{(p_u p_b)(p_d p_t)}{\left(\hat{s} - m_W^2\right)^2 + \Gamma_W^2 m_W^2}, \qquad |M|^2 = V_{tb}^2 V_{ud}^2 (g_W)^4 \frac{\hat{t}(\hat{t} - M_t^2)}{\left(\hat{s} - m_W^2\right)^2 + \Gamma_W^2 m_W^2}.$$

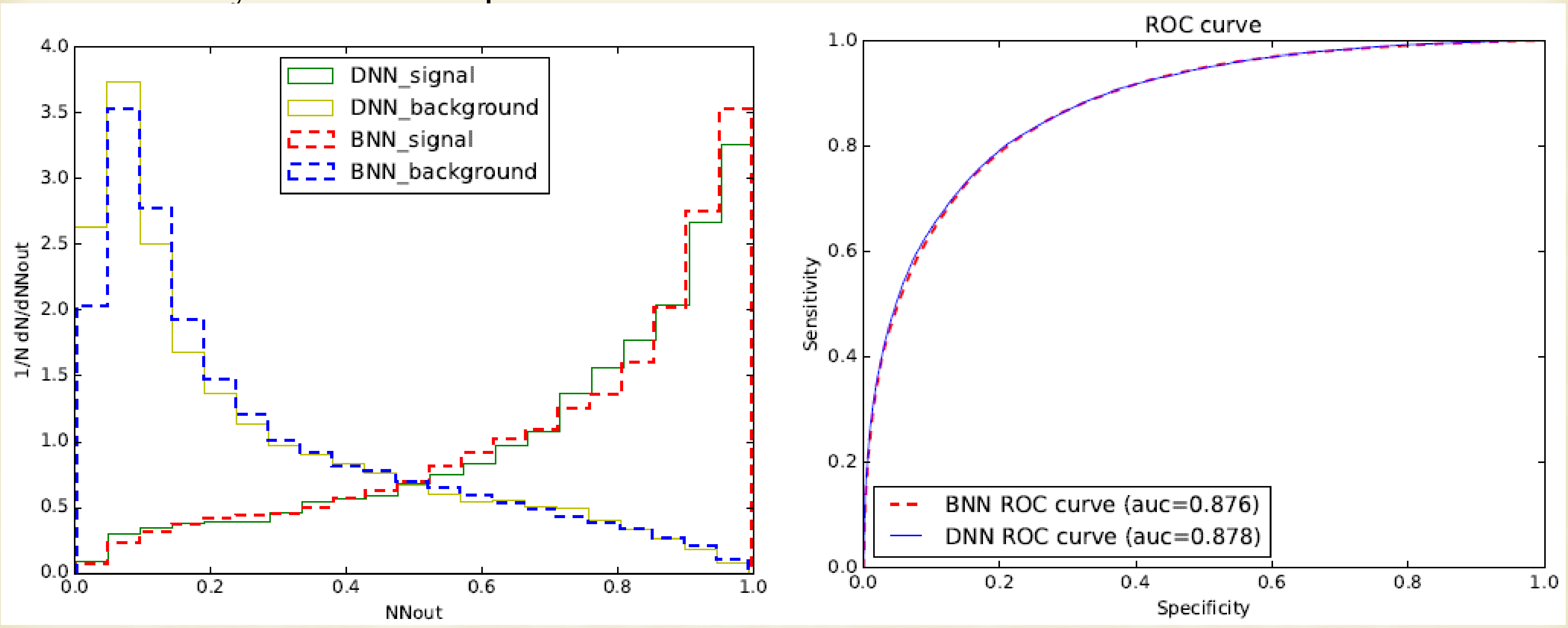
- Comparison of benchmark BNN with DNN trained on scalar products of final particles and s-Mandelstam variables. Eff. is still far from benchmark BNN.



Initial state particles 4-momenta as an input for DNN

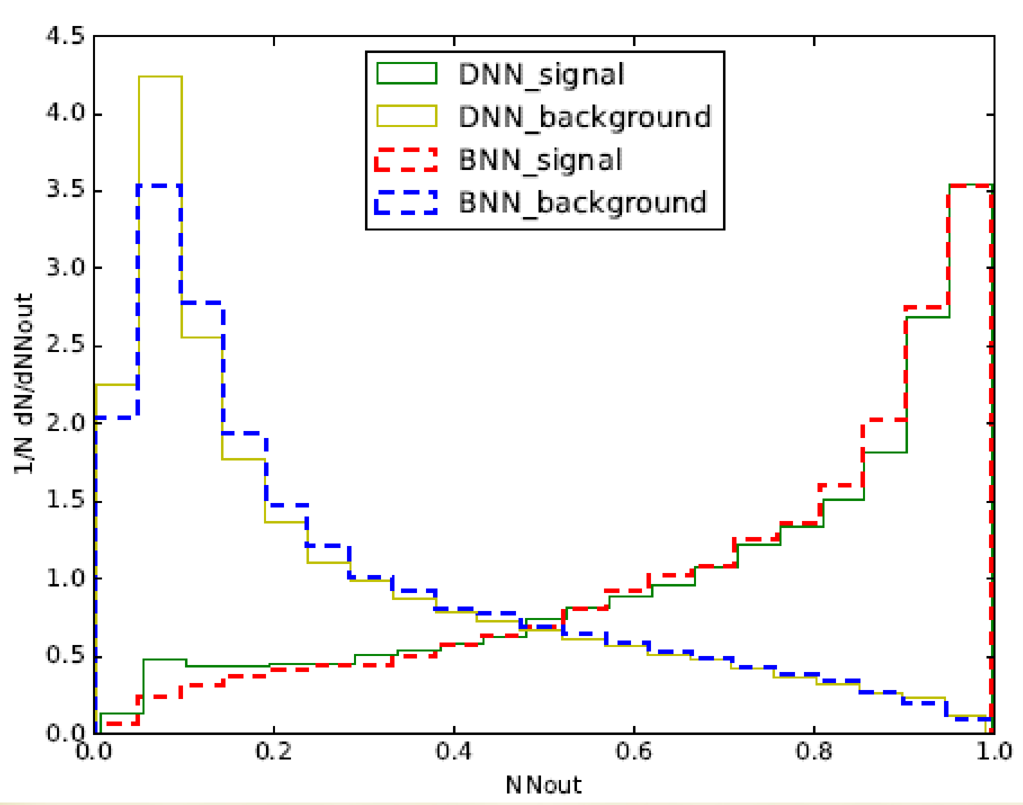
- M^2 is a function of not only final particles momenta, but initial particles momenta as well (not available for hadron colliders). In the massless case p_{in} can be represented as and approximated with P_T and pseudorapidity $\hat{t}_{i,f} = -\sqrt{\hat{s}} e^{Y} p_T^f e^{-|y_f|}$
[Phys.Atom.Nucl. 71 (2008) 2, 388-393]

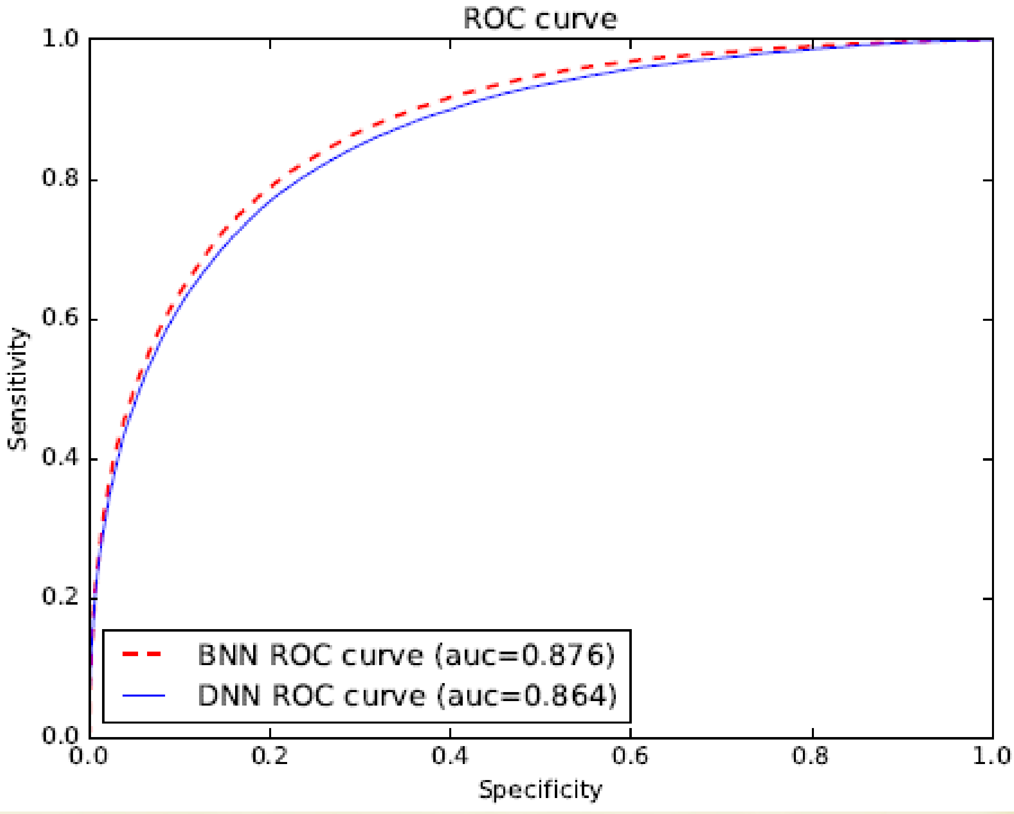
- General recipe to form input space for DNN analysis of the collider hard processes: take scalar products of 4-momenta of the final particles, Mandelstam variables (s,t,u), transverse momenta and pseudorapidity. ROC curve demonstrates desired efficiency with the recipe



Check for the completeness

- In additional to the proposed set one can add 4-momenta of the final particles.
- The comparison of benchmark BNN with DNN trained on the scalar-products of four-momenta, four-momenta and transverse momenta of the final particles and Mandelstam variables as the set of input variables. DNN has five layers.
- ROC curve demonstrates the completeness of the recipe, 4-momenta does not add more information, but increase the dimensionality which leads to more difficult training.





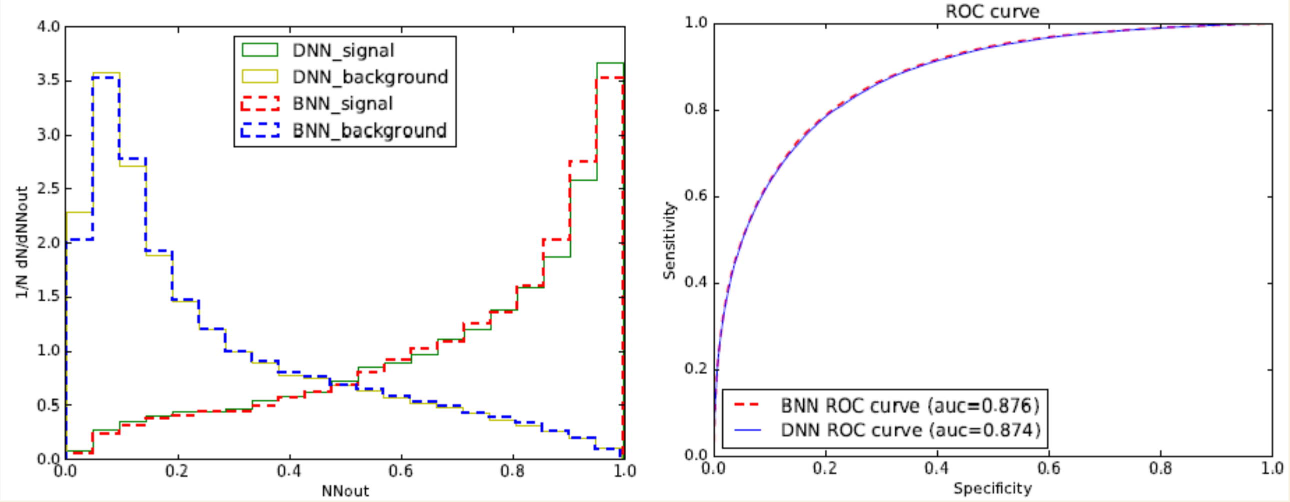
Conclusion

- The recommended recipe to form DNN input space to distinguish one hard scattering process from others includes three classes:
  - scalar products of 4-momenta of the final particles,
  - Mandelstam variables (only **s** are available for pp; **t**,**u** for lepton colliders),
  - transverse momenta and pseudorapidity of the final particles (to approximate t-channel Mandelstam variables which depends on initial particles momenta).
- The proposed set of raw observables covers the kinematic differences in hard processes. In additional, it is possible to add some other type of information (e.g. b-tagging discriminant, charge of lepton, …)
- More details can be found in Int.J.Mod.Phys.A 35 (2020) 21, 2050119 (for DNN), Phys.Atom.Nucl. 71 (2008) 2, 388-393 (for NN)

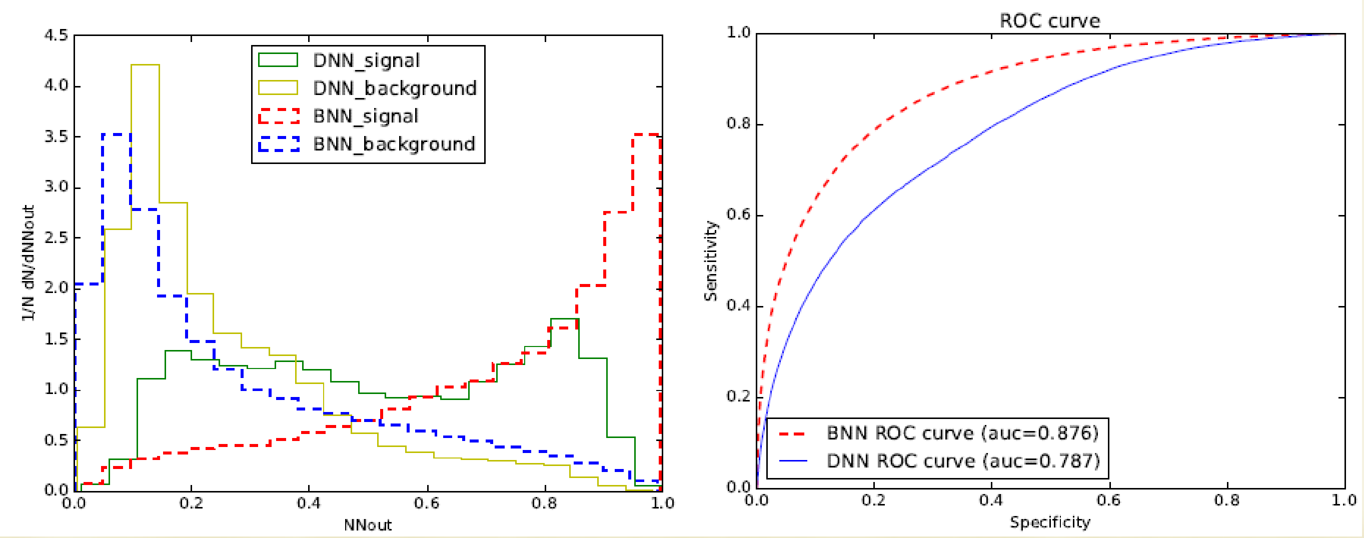
Back Up

cross checks

- Comparison of the Bayesian NN (BNN) and Deep Learning NN (DNN) methods with the same highly optimized benchmark set of variables:

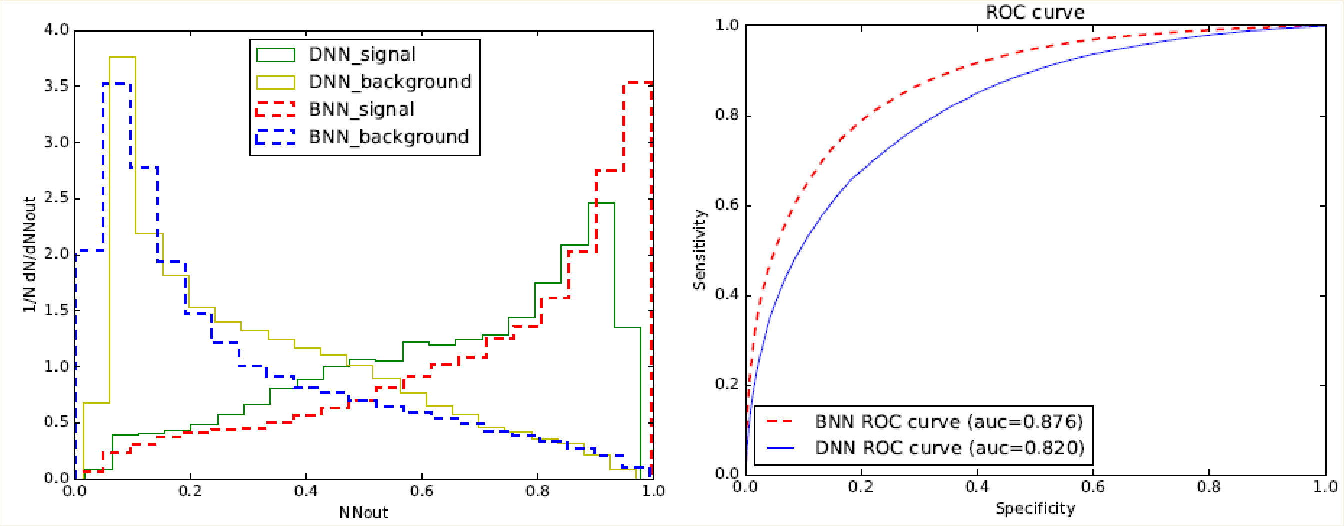


- Comparison of benchmark set and set with only scalar products of 4-momenta of final particles.

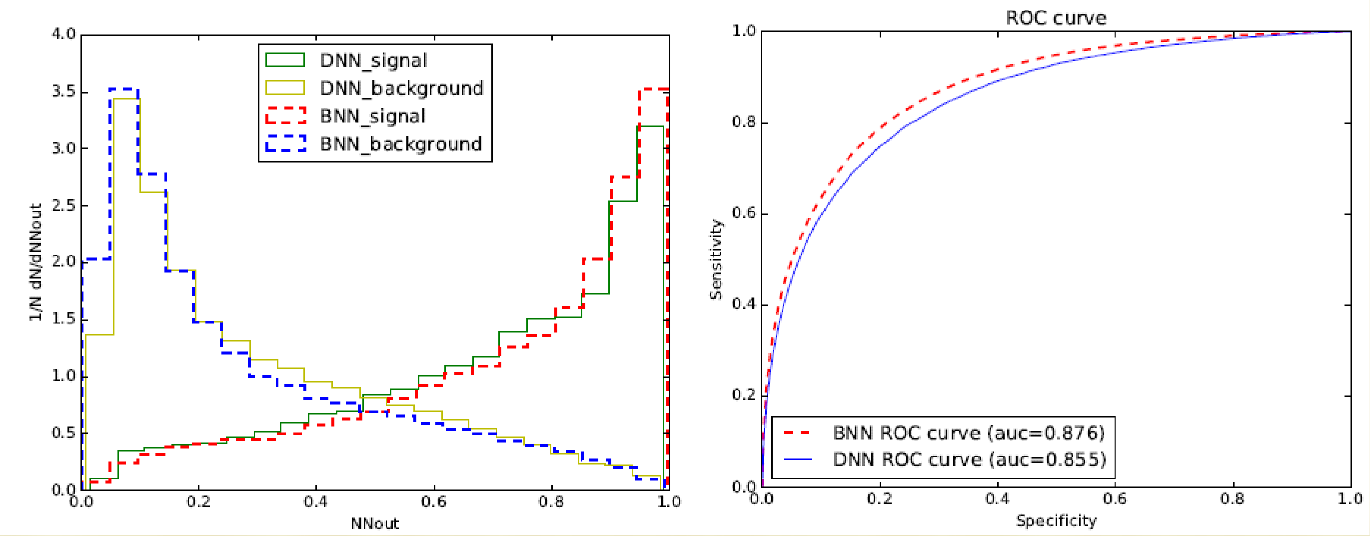


cross checks

The comparison of benchmark BNN with DNN trained on the scalar-products of four-momenta of the final particles and s-channel Mandelstam variables as the set of input variables. DNN has five hidden layers. The left plot demonstrates outputs of DNN and BNN for the signal and background processes. The ROC curves are shown in the right plot.

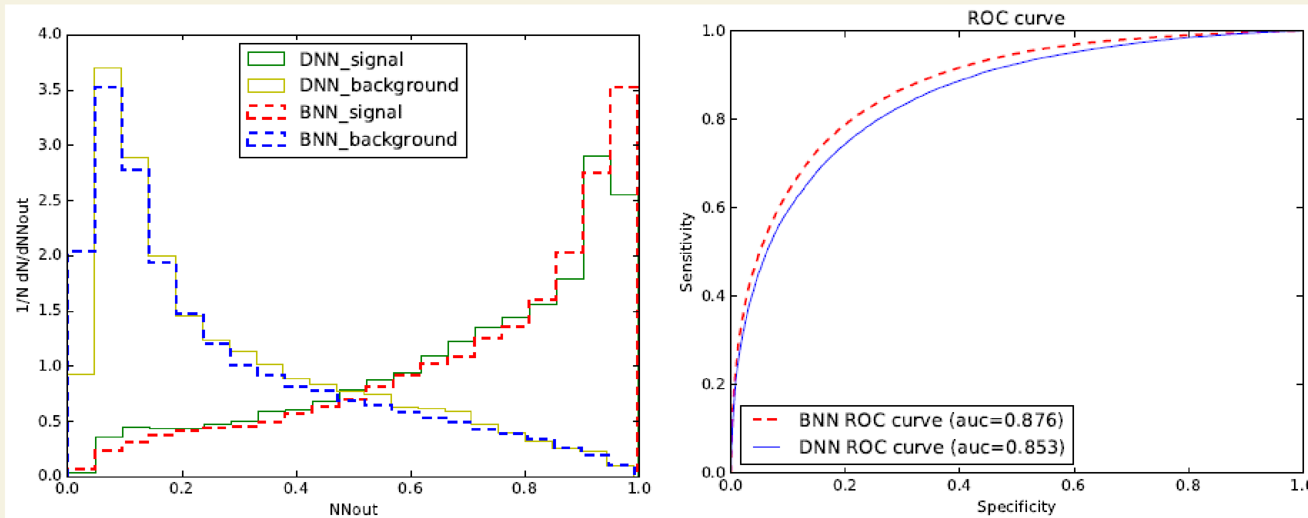


The comparison of benchmark BNN with DNN trained on the scalar-products of four-momenta and four-momenta of the final particles as the set of input variables. DNN has five hidden layers.



cross checks

The comparison of benchmark BNN with DNN trained on the scalar-products of four-momenta, four-momenta of the final particles and s-channel Mandelstam variables as the set of input variables. DNN has five hidden layers. The left plot demonstrates outputs of DNN and BNN for the signal and background processes. The ROC curves are shown in the right plot.



The comparison of benchmark BNN with DNN trained on the scalar-products of four-momenta, transverse momenta of the final particles and Mandelstam variables as the set of input variables. DNN has three hidden layers.