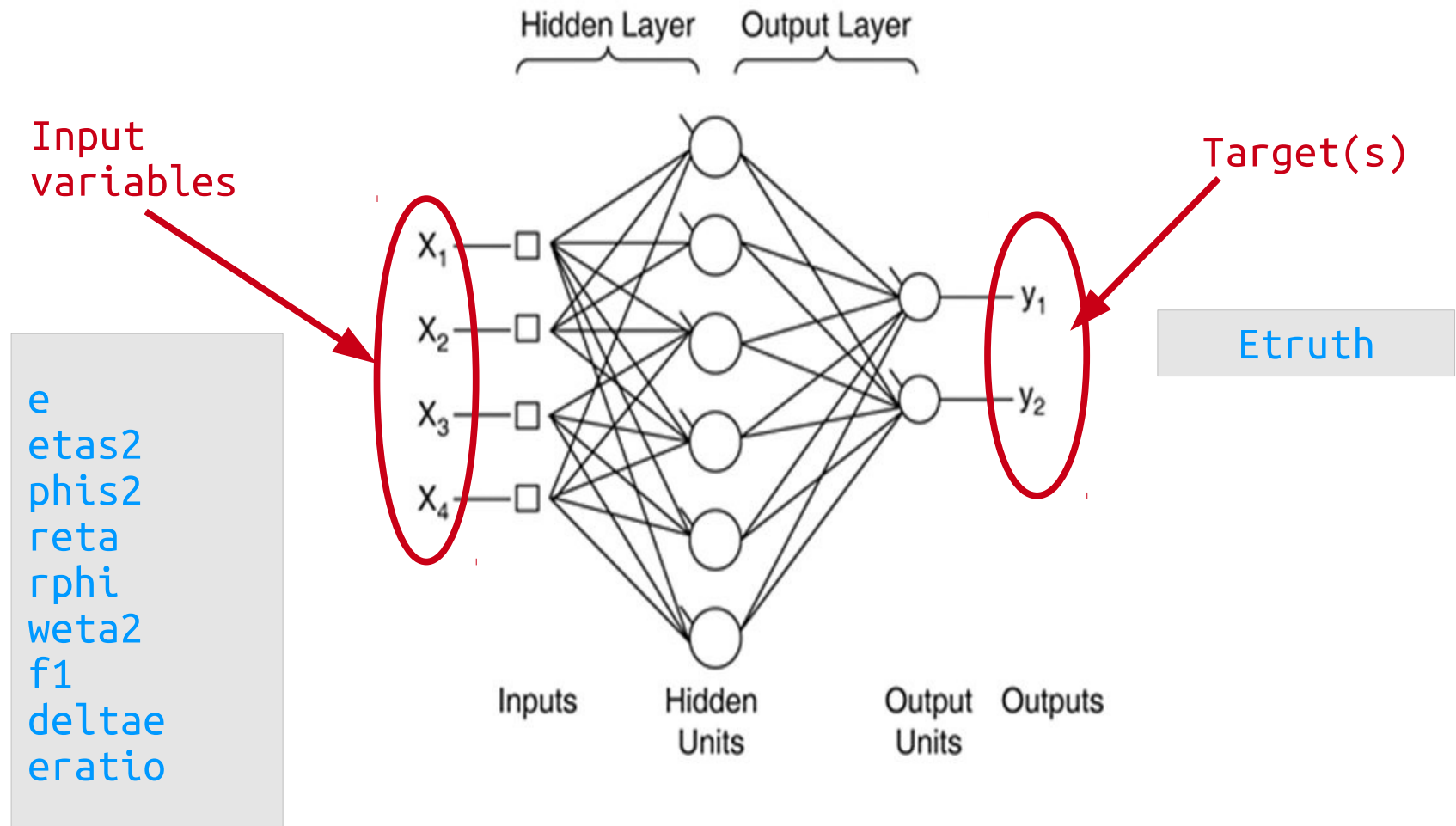

MVA regression analysis for electrons and
photons energy correction

S. Bordoni, B. Laforge, L. Yao

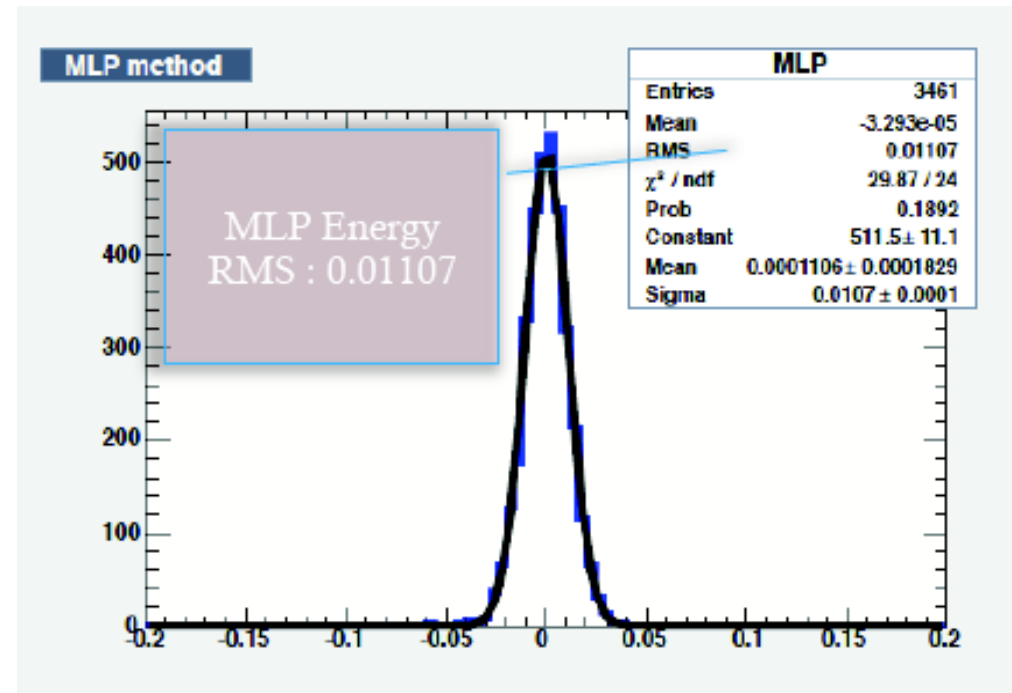
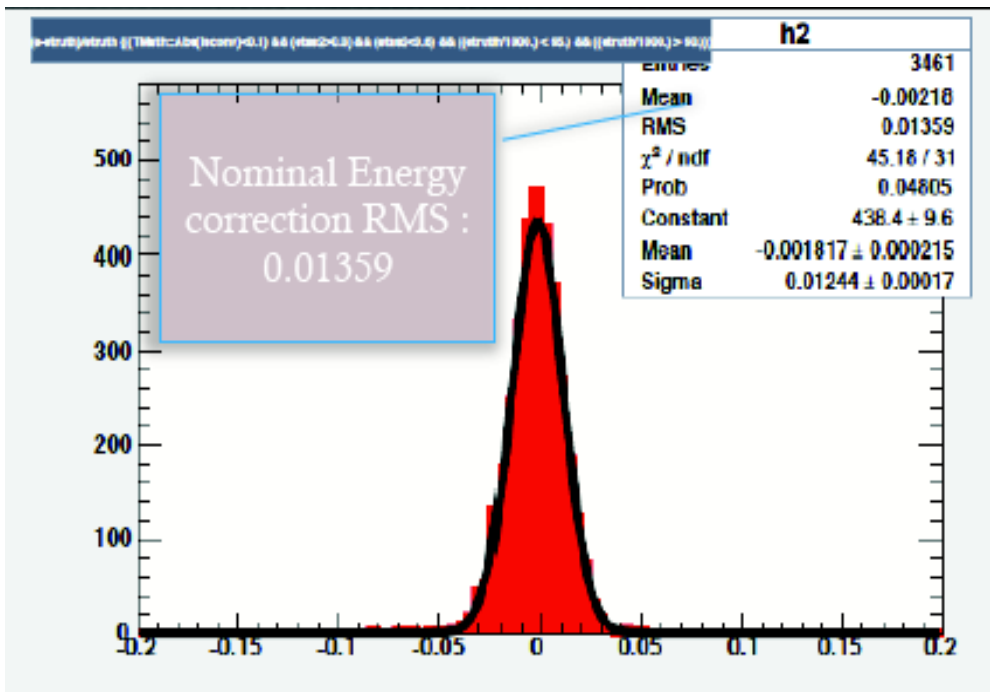
4 May 2012

MultiLayer Perceptron

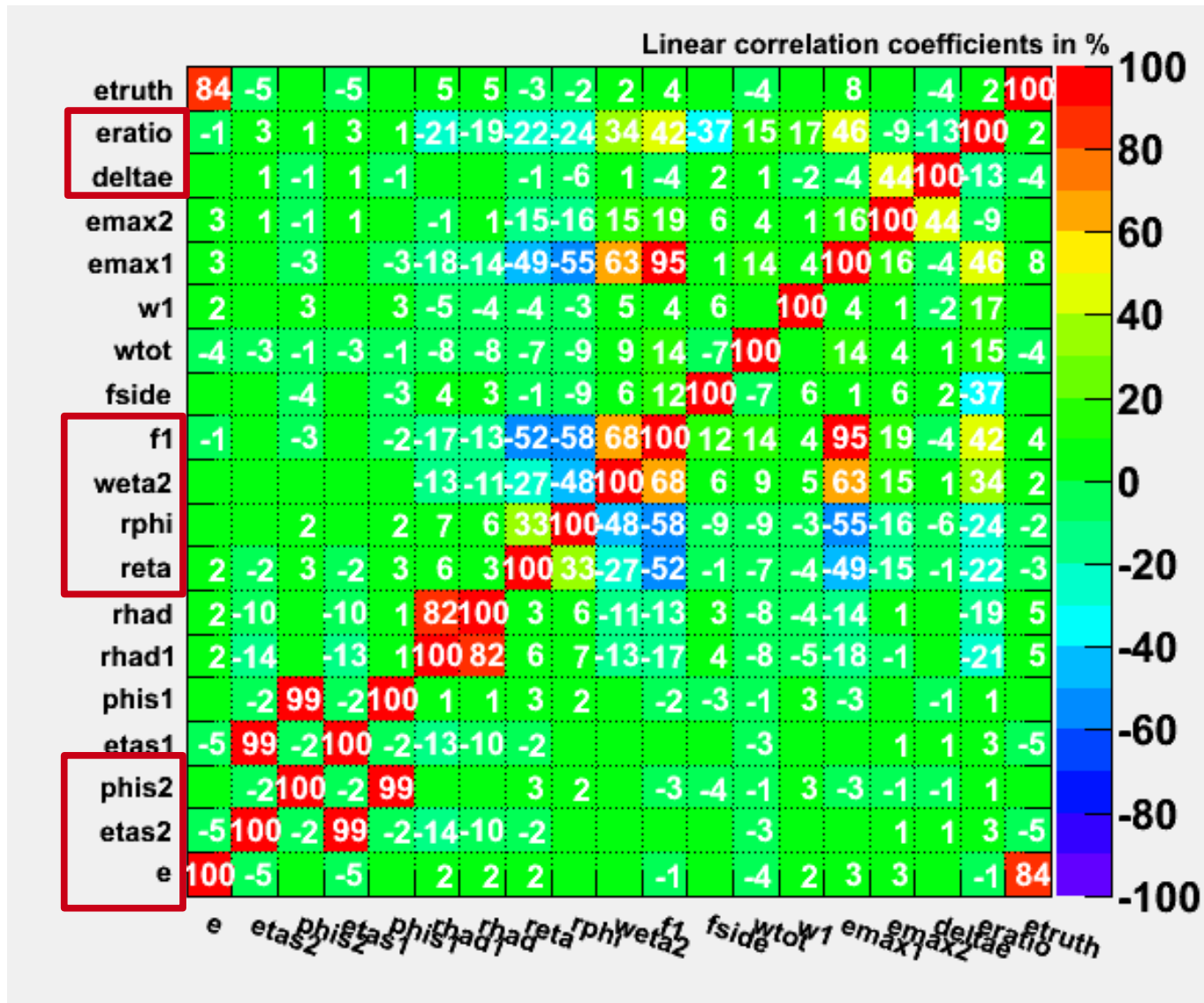


Previous results

Promising results shown by Liwen in April



Correlation Matrix



Study of the MLP neural network

The configuration of the MLP network can be modified by the user

```
// Neural network (MLP)
if (Use["MLP"])
  factory->BookMethod( TMVA::Types::kMLP, "MLP", "!H:!V:VarTransform=Norm:
NeuronType=tanh:NCycles=20000:HiddenLayers=N+20:
TestRate=6:TrainingMethod=BFGS:Sampling=0.3:SamplingEpoch=0.8
:ConvergenceImprove=1e-6:ConvergenceTests=15:!UseRegulator" );
```

More options are available! <http://tmva.sourceforge.net/optionRef.html#MVA::MLP>

In the following slides:

Study the response of the MLP neural network (for a fixed set of input variables), as function of:

- number of neurons on the Hidden Layer
- number of Hidden layers
- statistics available (training/test sample)

Setup

Photon Selection

- Unconverted
- Energy in [60,65] GeV
- η in [0.3, 0.6]

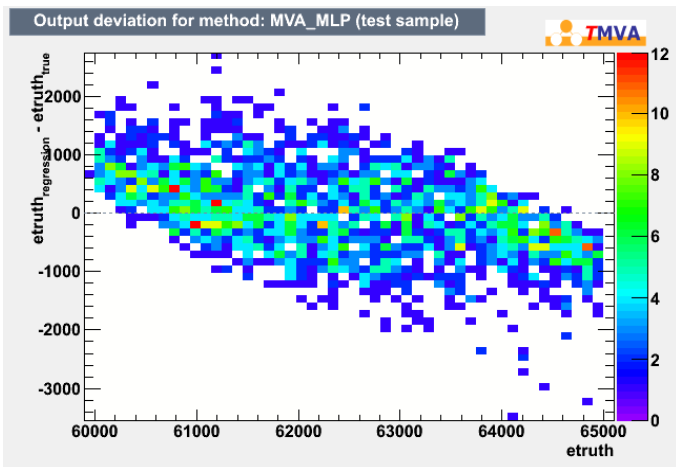
MLP setup

- Input: subset of shower shape variables and η, ϕ position on the s_2
- Target: `etruth`

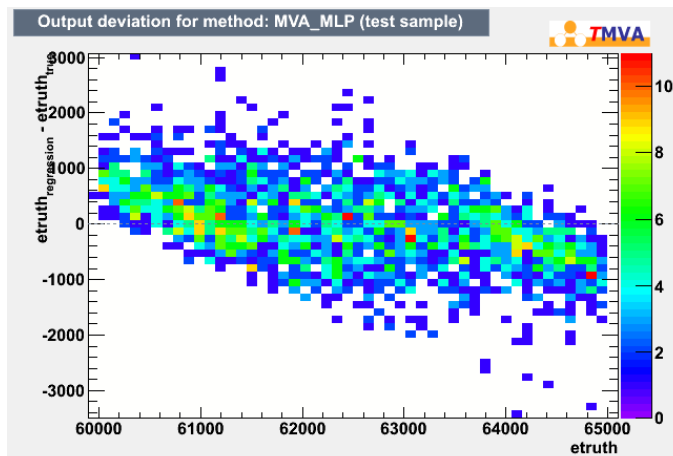
- Training samples: background events (complete list in backup slides)
- Test sample: ggH @ 120 GeV
- Other samples: ggH @ 100-150 GeV
 - used to vary the statistics of the training/test sample

Linearity tests

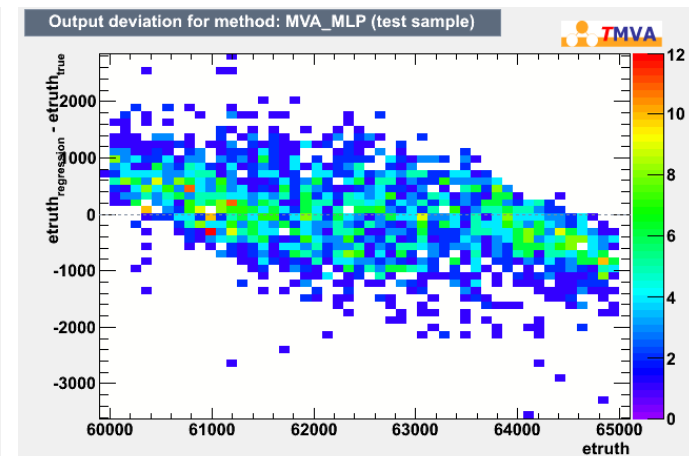
HiddenLayer N+5



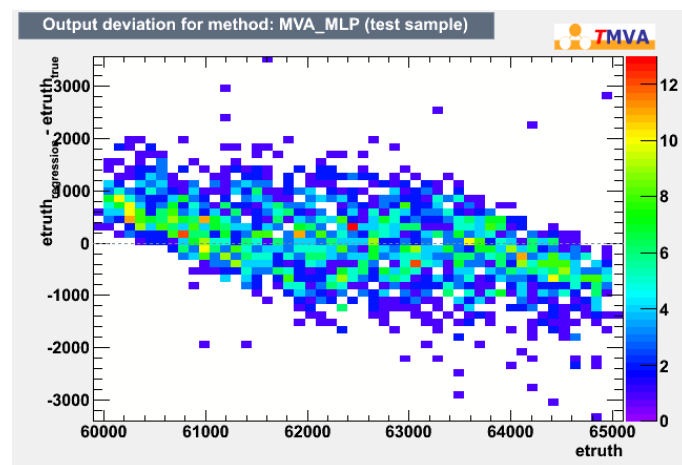
HiddenLayer N+10



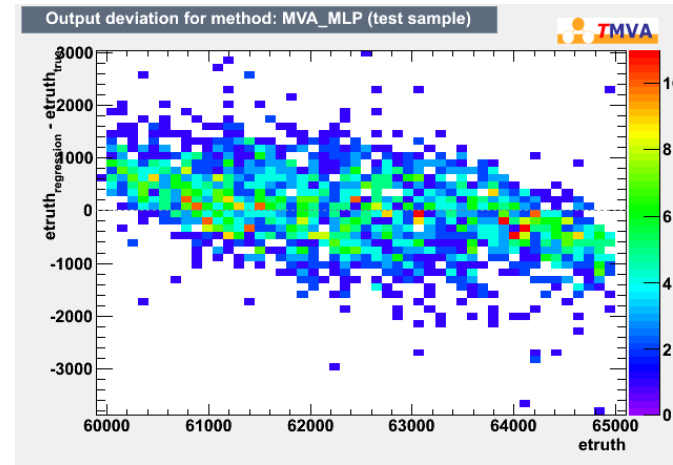
HiddenLayer N+15



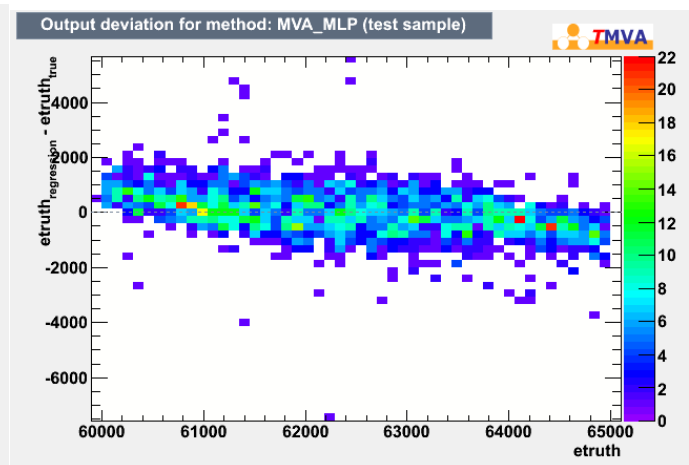
HiddenLayer N+20



HiddenLayer N+50



HiddenLayer N+20, N+40



Some results

- Variation of the number of neurons in the Hidden Layer

	N	N+5	N+10	N+20	N+50	N+20,N+50
Nominal correction	0.01359	0.01359	0.01359	0.01359	0.01359	0.01359
MLP correction	0.01124	0.01129	0.01149	0.0126	0.01275	0.1375
%	17.3	16.9	15.5	7.3	6.2	

$$\% = (\text{Nominal correction} - \text{MLP correction}) / \text{Nominal correction}$$

Some results (2)

- Adding statistics on the **training** sample (HL = N+20)

# photons (train)	# photons (test)	MLP correction	Nominal correction	%
2405	3461	0.0126x	0.01359	7.3
3430	3461	0.01173	0.01359	13.7
4859	3461	0.01165	0.01359	14.3
6310	3461	0.01106	0.01359	18.6

$$\% = (\text{Nominal correction} - \text{MLP correction}) / \text{Nominal correction}$$

Some results (3)

- Adding statistics on the **test** sample (HL = N+20)

# photons (train)	# photons (test)	MLP correction	Nominal correction	%
2405	3461	0.0126	0.01359	7.2%
2405	5820	0.01269	0.01376	7.8%
2405	10027	0.01265	0.01359	6.9%
2405	12423	0.01279	0.01366	6.4%

$$\% = (\text{Nominal correction} - \text{MLP correction}) / \text{Nominal correction}$$

Conclusions

- For a fixed set of input variables and a fixed statistics (train & test sample) the MLP NN seems to be more stable with a small number of neurons in the hidden layer, whereas the linearity of the energy correction improves with an increasing number of neurons in the hidden layer.
- Large statistics for the training test is necessary to have a good working point of the neural network.
- A big discrepancy on the statistics for the training and the test sample damage the results.

To Do:

- Use the information of the vertex position in input
- Vary the input variables (number and type)
- Use the information given by the Tool about the training performances
- Put the code on svn

Backup

Samples

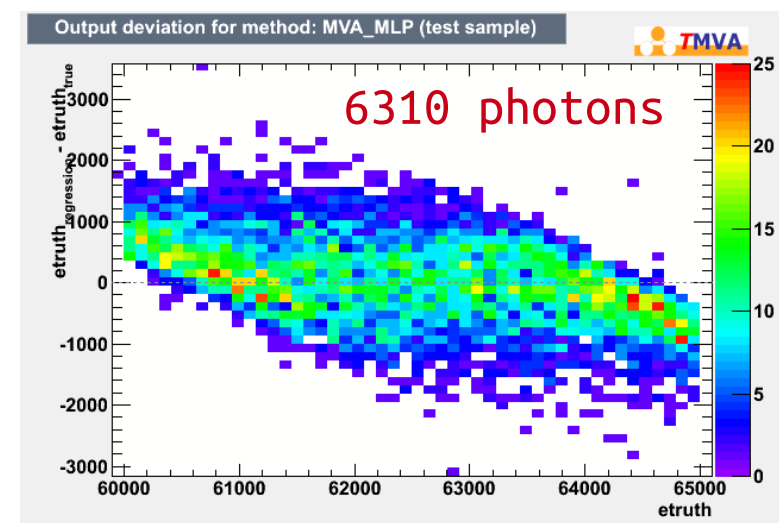
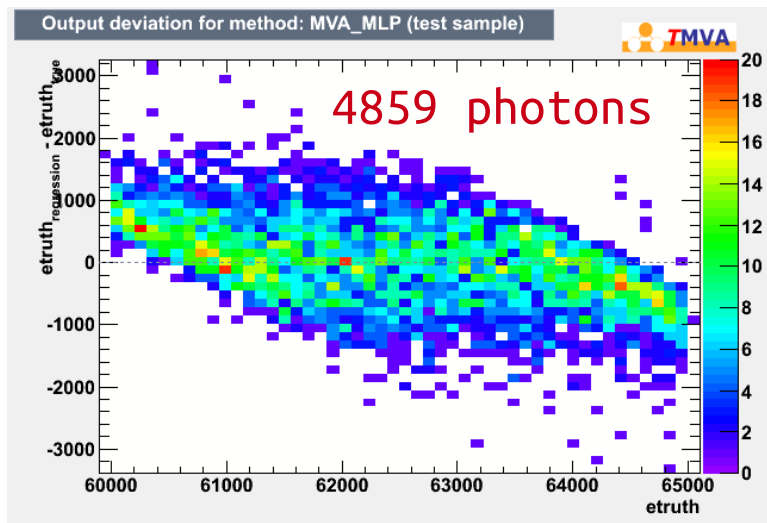
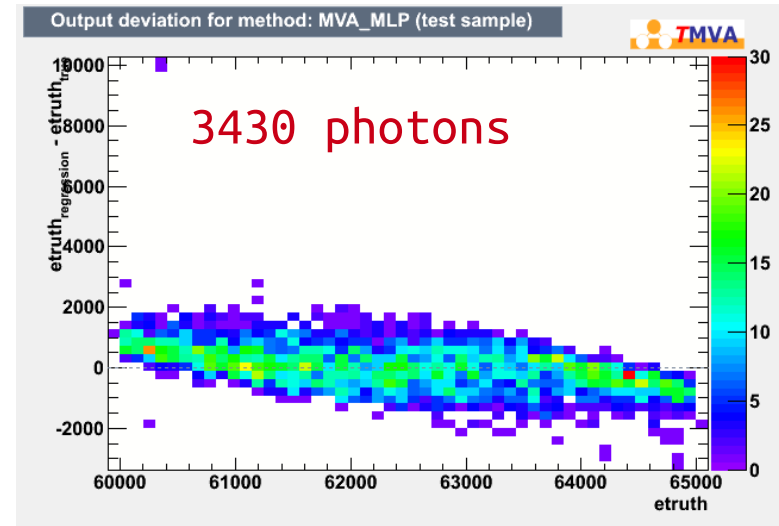
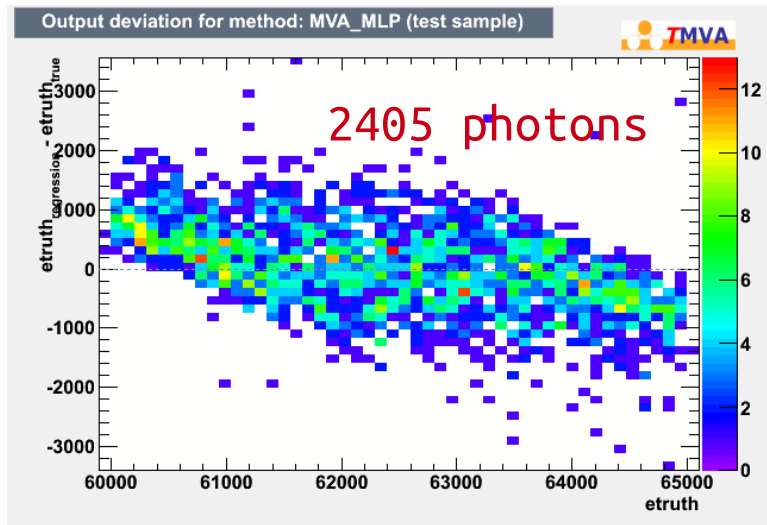
Training:

mc11_7TeV.105964.PythiaGamgam15.merge.NTUP_PHOTON.e939_s1310_s1300_r3044_r2993_p868_tid686696_00
mc11_7TeV.113714.SherpaY4JetsPt35.merge.NTUP_PHOTON.e972_s1310_s1300_r3043_r2993_p868_tid686691_00
mc11_7TeV.113715.SherpaY4JetsPt70.merge.NTUP_PHOTON.e972_s1310_s1300_r3043_r2993_p868_tid686692_00
mc11_7TeV.113716.SherpaY4JetsPt140.merge.NTUP_PHOTON.e972_s1310_s1300_r3043_r2993_p868_tid686693_00
mc11_7TeV.113717.SherpaY4JetsPt280.merge.NTUP_PHOTON.e972_s1310_s1300_r3043_r2993_p868_tid686694_00
mc11_7TeV.119081.2DP20_GamJetGamGam_pythia_photon_filter.merge.NTUP_PHOTON.e825_s1310_s1300_r2935_r2900_p868
mc11_7TeV.119081.2DP20_GamJetGamGam_pythia_photon_filter.merge.NTUP_PHOTON.e825_s1310_s1300_r3044_r2993_p868
mc11_7TeV.119081.2DP20_GamJetGamGam_pythia_photon_filter.merge.NTUP_PHOTON.e825_s1356_s1353_r2935_r2900_p868
mc11_7TeV.126371.SherpaY4JetsPt500.merge.NTUP_PHOTON.e972_s1310_s1300_r3043_r2993_p868_tid686695_00
mc11_7TeV.126372.SherpaY4JetsPt15.merge.NTUP_PHOTON.e972_s1310_s1300_r3043_r2993_p868_tid686690_00
mc11_7TeV.126389.Sherpa2DP20.merge.NTUP_PHOTON.e1028_s1310_s1300_r3108_r3109_p868

Application:

mc11_7TeV.116610.PowHegPythia_ggH120_gamgam.merge.NTUP_PHOTON.e873_s1310_s1300_r3043_r2993_p868_tid693293_00
mc11_7TeV.116610.PowHegPythia_ggH120_gamgam.merge.NTUP_PHOTON.e873_s1310_s1300_r3043_r2993_p868_tid693294_00

Linearity tests vs statistics of the training sample



Some results (from TMVA training output)

	N	N + 5	N + 10	N + 20	N + 50	N + 20, N + 40
<u>Training Sample</u>						
bias	-14.5	-12.0	-22.8	-16.6	-8.32	-23.9
bias_T	-13.7	-8.41	-21.8	-7.8	-2.78	-17.9
RMS	708	710	719	735	743	819
RMS_T	617	618	627	633	640	676
<u>Test sample</u>						
bias	-0.417	-1.03	1.11	-1.88	-4.48	-0.965
bias_T	1.57	6.96	10.1	1.79	1.48	9.38
RMS	676	678	668	658	646	615
RMS_T	588	593	587	578	565	539