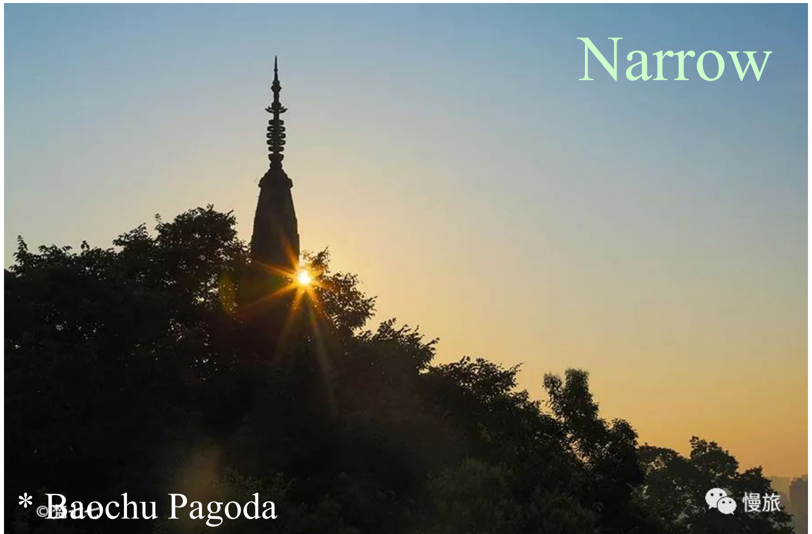
Learning a broad resonance at the LHC

Ke-Pan Xie (谢柯盼)
Seoul National University

Jan 24 2019, @Sun Yat-sen University, Guangzhou

In collaboration of Sunghoon Jung and Dongsub Lee
Paper in preparation

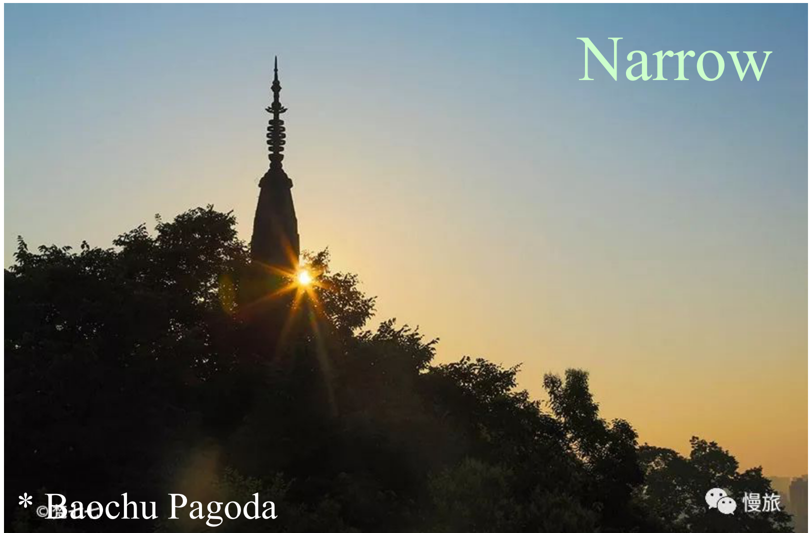
➢ The resonances of new physics



W', *Z'* in the weak gauge extension models;
H^0, $H^{\pm}$, A in 2HDM, etc.

KK gluons,
composite Higgs resonances,
etc.

➢ The resonances of new physics



W', *Z'* in the weak gauge extension models;
H^0, $H^{\pm}$, A in 2HDM, etc.

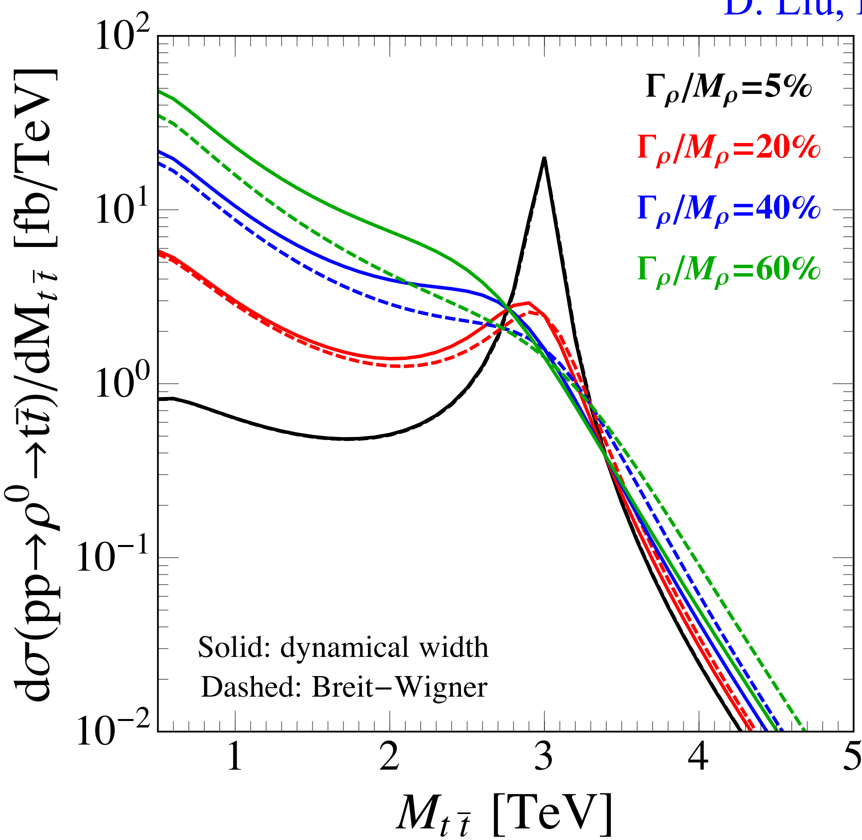
For an interaction $g_\rho t\gamma^\mu \bar{t}\rho_\mu$,

$$\frac{\Gamma_{\rho\to t\bar{t}}}{M_\rho} = \frac{g_\rho^2}{4\pi}.$$

KK gluons,
composite Higgs resonances,
etc.

➢ The invariant mass of a resonance

D. Liu, L.-T. Wang and K.-P. Xie, arXiv:1901.01674



$$\frac{1}{(\hat{s}-M_\rho^2)^2+M_\rho^2\Gamma_\rho^2}$$

Breit-Wigner

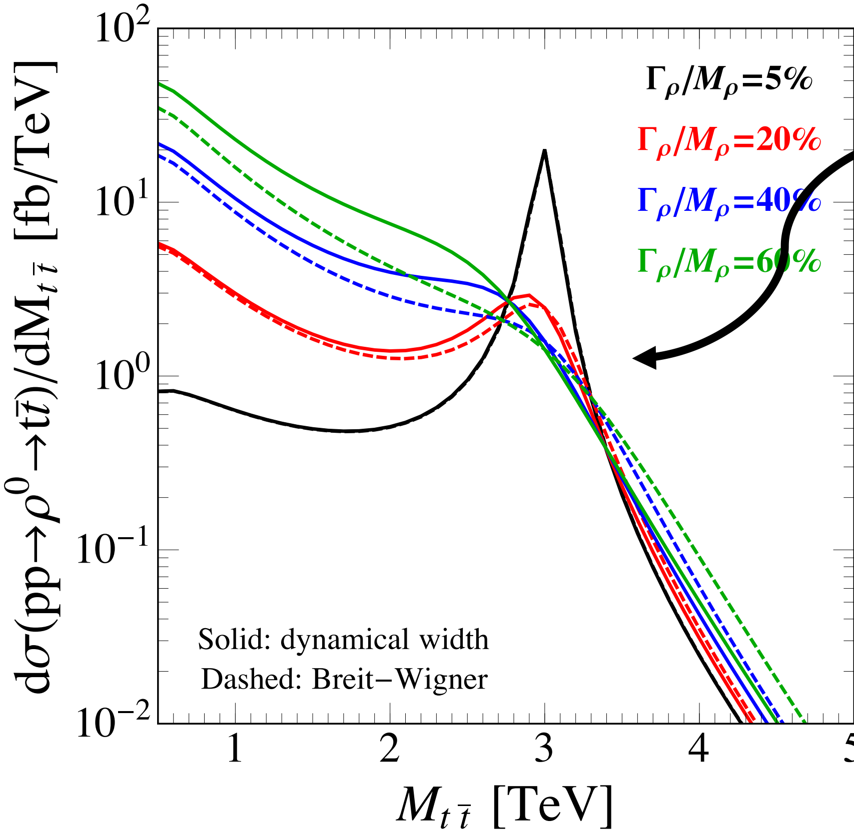
v.s.

dynamical width

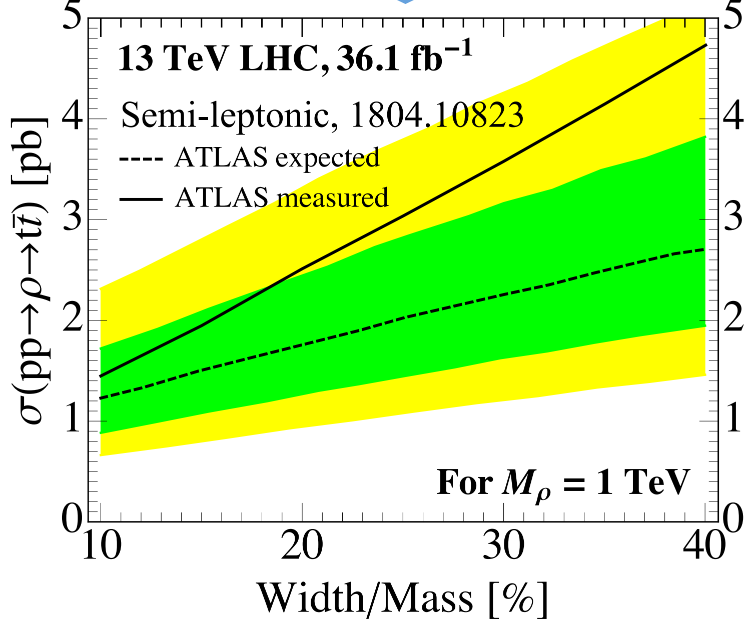
$$\frac{1}{(\hat{s}-M_\rho^2)^2+\hat{s}^2\Gamma_\rho^2/M_\rho^2}$$

– Breit-Wigner distribution is commonly used, but a more suitable treatment is the dynamical width approach!

➢ Searching for a *ttbar* resonance

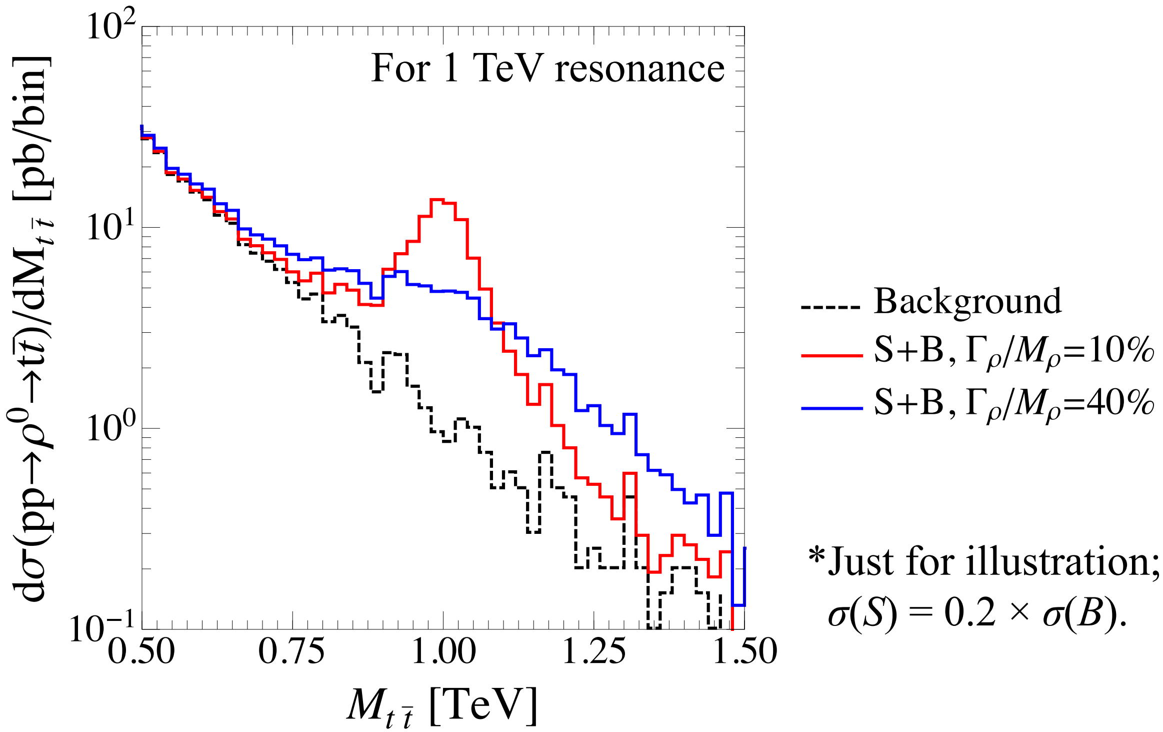


The traditional approach: fit M_{tt} distribution and get the cross section limit:



– The traditional method becomes worse when the width of the resonance increases!

– This worse result is expected, because the traditional approach makes use of only the invariant mass;

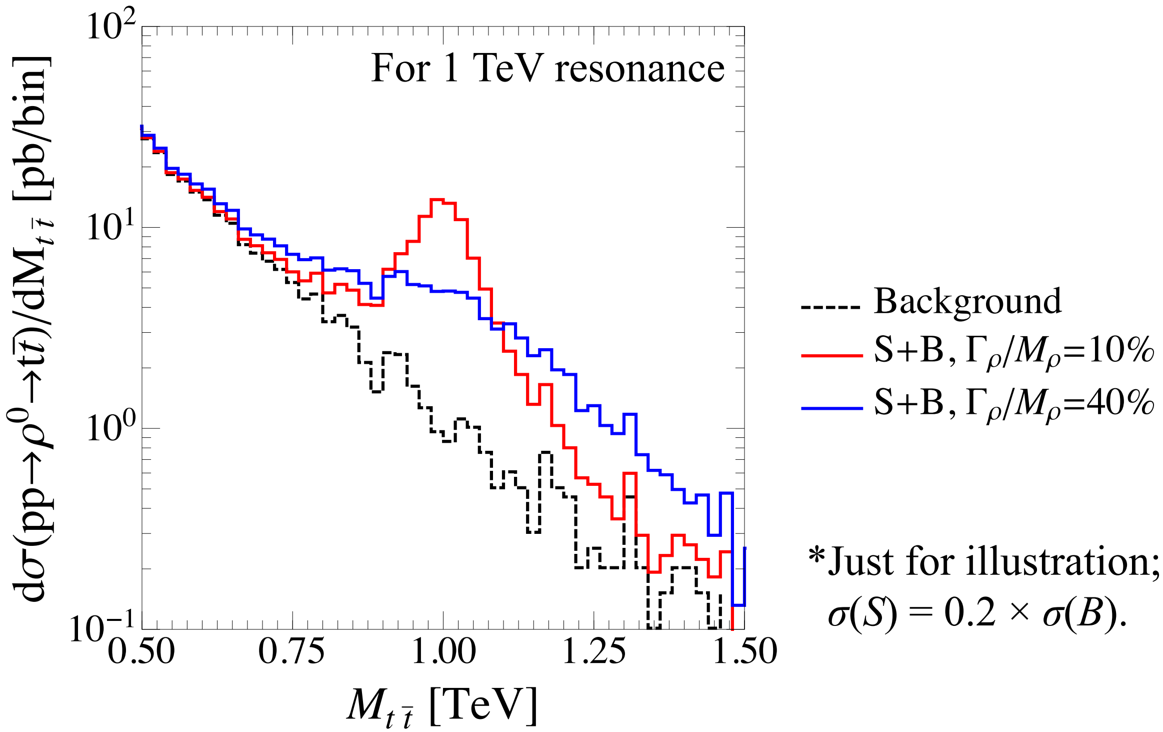


*Just for illustration; assume $\sigma(S) = 0.2 \times \sigma(B)$.

– However, the invariant mass peak becomes less prominent while the width increases!

➢ Searching for a *ttbar* resonance using DNN

– We try to use Deep Neural Network to make use of all the observables in the final state;



*Just for illustration; assume $\sigma(S) = 0.2 \times \sigma(B)$.

– We expect the efficiency at large width region can be improved.

– The process considered in this work:

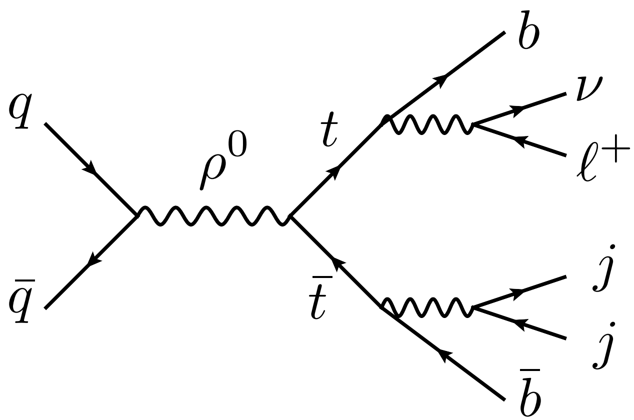
$$\text{Signal}: \; pp \to \rho^0 \to t\bar{t} \to 1\ell^\pm + \text{jets}$$

$$\text{Background}: \; \text{SM } pp \to t\bar{t} \to 1\ell^\pm + \text{jets}$$

– The benchmarks:

1. Mass $M_\rho = 1$ TeV,
   width $\Gamma_\rho/M_\rho = 10\%$, 20%, 30% and 40%;
   denoted as signal models M1Γ1 ~ M1Γ4.
2. Mass $M_\rho = 5$ TeV,
   width $\Gamma_\rho/M_\rho = 10\%$, 20%, 30% and 40%;
   denoted as signal models M5Γ1 ~ M5Γ4.

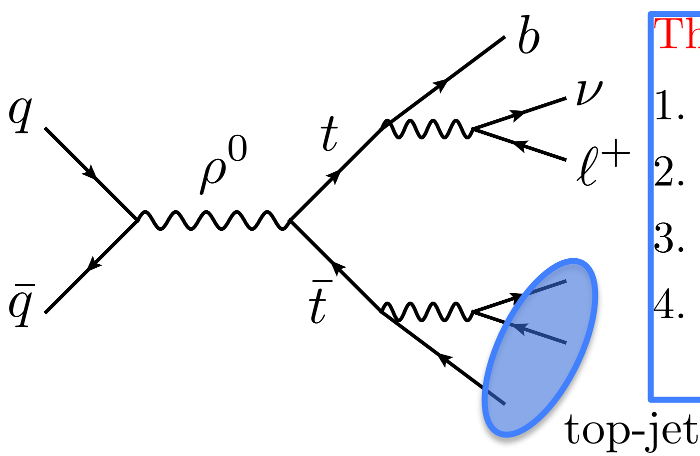
– We define 2 kinematic regions:



For models M1Γ1 ~ M1Γ4

The resolved region:

1. $1\ell^{\pm}$ with $p_T > 30$ GeV and $|\eta| < 2.5$;
2. $\not{E}_T > 20$ GeV and $\not{E}_T + M_T > 60$ GeV;
3. 4 jets with $p_T > 25$ GeV and $|\eta| < 2.5$;



For models M1Γ1 ~ M1Γ4 and M5Γ1 ~ M5Γ4

The boosted region:

1. $1\ell^{\pm}$ with $p_T > 30$ GeV and $|\eta| < 2.5$;
2. $\not{E}_T > 20$ GeV and $\not{E}_T + M_T > 60$ GeV;
3. 1 top jet with $p_T > 300$ GeV and $|\eta| < 2.0$;
4. 1 selected jet with $p_T > 25$ GeV and $|\eta| < 2.5$ and $\Delta R(j, \ell) < 1.5$.

– In Total we have 4 + 8 = 12 benchmark signal models

– And then we generate events:

Resolved region:

Process	Event number	Cut 1	Cut 2	Cut 3	Efficiency
M1Γ1	5.00×10^6	3.32×10^6	3.02×10^6	1.81×10^6	36.3%
M1Γ2	5.00×10^6	3.29×10^6	2.98×10^6	1.79×10^6	35.8%
M1Γ3	3.85×10^6	2.52×10^6	2.23×10^6	1.36×10^6	35.3%
M1Γ4	5.00×10^6	3.25×10^6	2.93×10^6	1.75×10^6	34.9%
SM $t\bar{t}$	4.98×10^6	2.60×10^6	2.21×10^6	1.39×10^6	28.0%

SM background cross section after cuts: 68.9 pb

Boosted region:

Process	Event number	Cut 1	Cut 2	Cut 3	Cut 4	Efficiency
M1Γ1	5.00×10^6	3.32×10^6	3.02×10^6	1.17×10^6	9.61×10^5	19.2%
M1Γ2	5.00×10^6	3.29×10^6	2.98×10^6	1.08×10^6	8.85×10^5	17.7%
M1Γ3	5.00×10^6	3.27×10^6	2.96×10^6	1.02×10^6	8.34×10^5	16.7%
M1Γ4	5.05×10^6	3.28×10^6	2.96×10^6	9.92×10^5	8.06×10^5	15.9%
M5Γ1	5.00×10^6	2.53×10^6	2.36×10^6	1.15×10^6	8.41×10^5	16.8%
M5Γ2	5.00×10^6	2.72×10^6	2.52×10^6	1.19×10^6	8.76×10^5	17.5%
M5Γ3	5.00×10^6	2.81×10^6	2.59×10^6	1.19×10^6	8.85×10^5	17.7%
M5Γ4	5.00×10^6	2.86×10^6	2.64×10^6	1.20×10^6	8.90×10^5	17.8%
SM $t\bar{t}$ (`xptj=150`)	1.99×10^7	1.22×10^7	1.08×10^7	1.41×10^6	1.21×10^6	6.10%

SM background cross section after cuts: 2.88 pb

- Now we take M1Γ4[$M\rho$ = 1 TeV, $\Gamma\rho/M\rho$ = 40%] as an example to describe the usage of DNN:
  - In resolved region, each event has 26 low-level observables:

1	2	3	4	5	6	7	8	9	10	11	12	13
E^{ℓ}	p_T^{ℓ}	η^{ℓ}	ϕ^{ℓ}	$\not{E}_T$	$\phi^{\not{E}_T}$	E^{j_1}	$p_T^{j_1}$	η^{j_1}	ϕ^{j_1}	b^{j_1}	E^{j_2}	$p_T^{j_2}$

14	15	16	17	18	19	20	21	22	23	24	25	26
η^{j_2}	ϕ^{j_2}	b^{j_2}	E^{j_3}	$p_T^{j_3}$	η^{j_3}	ϕ^{j_3}	b^{j_3}	E^{j_4}	$p_T^{j_4}$	η^{j_4}	ϕ^{j_4}	b^{j_4}

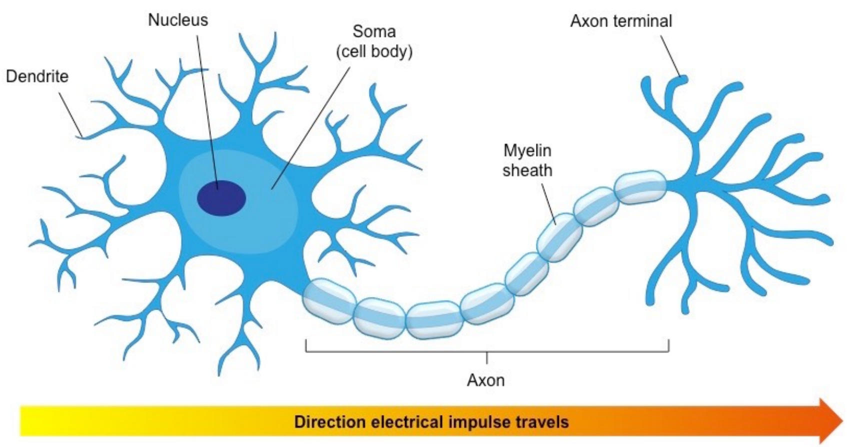
  and 1 label:

$$\text{signal (label 1)} \rightarrow \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad \text{background (label 0)} \rightarrow \begin{bmatrix} 1 \\ 0 \end{bmatrix}.$$

  - We mix 500,000 signal events and 500,000 background events to get a training dataset with size 1,000,000.

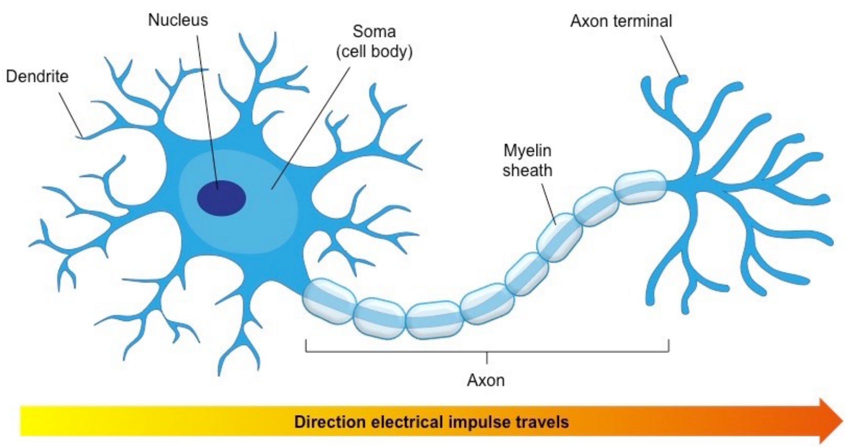
- DNN is a kind of machine learning technique (alternative: BDT[Boosted Decision Tree], SVM[Support Vector Machine], etc);
- It is vaguely inspired by the biological neural networks that constitute animal brains.[wikipedia]
- The basic unit of the DNN: **neuron**

- DNN is a kind of machine learning technique (alternative: BDT[Boosted Decision Tree], SVM[Support Vector Machine], etc);
- It is vaguely inspired by the biological neural networks that constitute animal brains.[wikipedia]
- The basic unit of the DNN: **neuron**

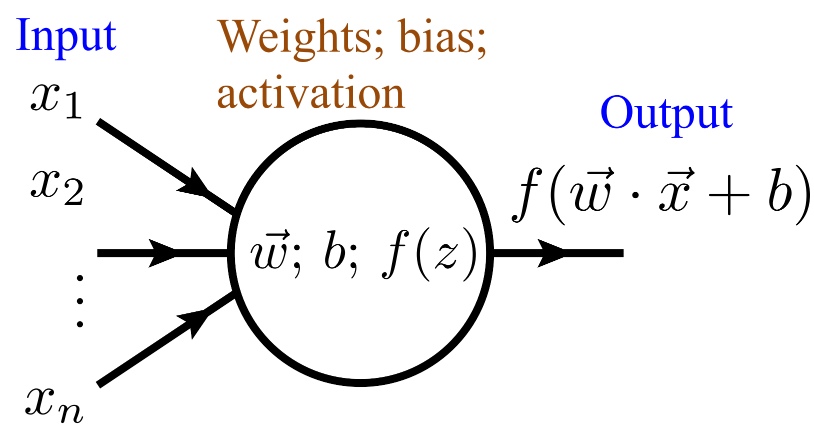


A boiological neuron

- DNN is a kind of machine learning technique (alternative: BDT[Boosted Decision Tree], SVM[Support Vector Machine], etc);
- It is vaguely inspired by the biological neural networks that constitute animal brains.[wikipedia]
- The basic unit of the DNN: **neuron**

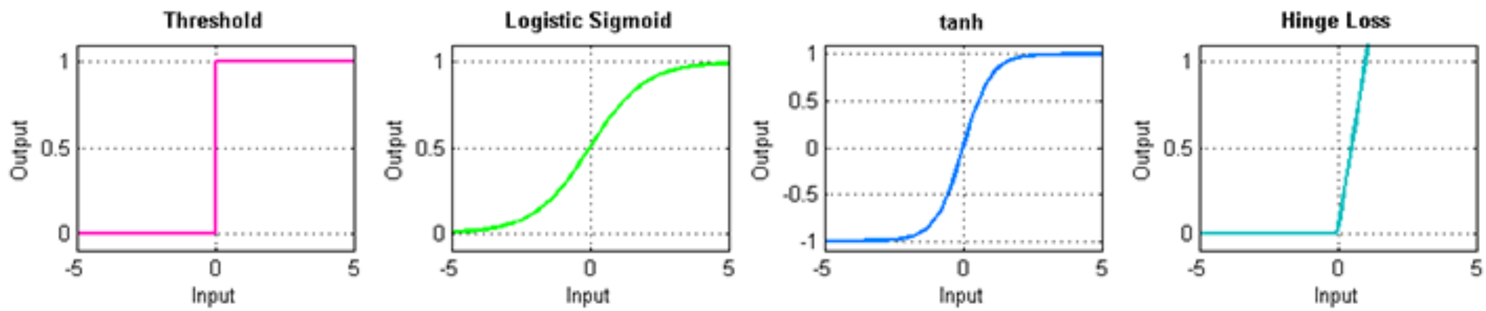


A boiological neuron



An artificial neuron

- DNN is a kind of machine learning technique
-

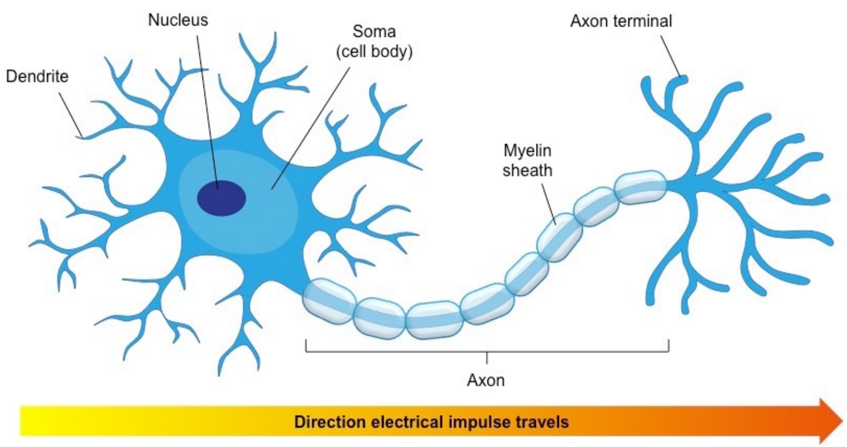


Undefined gradient
Cant be used

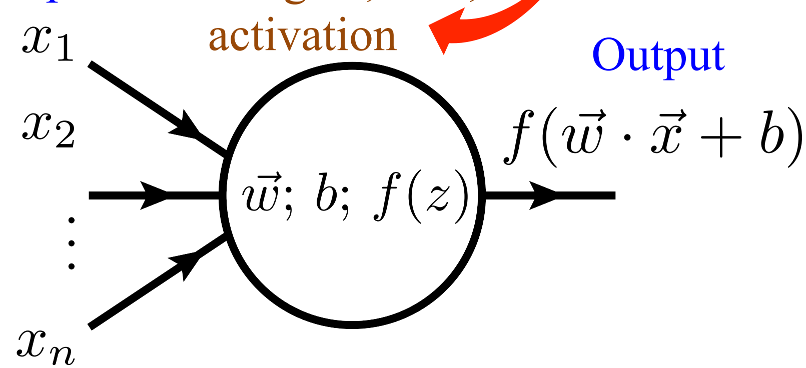
The common activation functions

New default activation

- The basic unit of the DNN: **neuron**

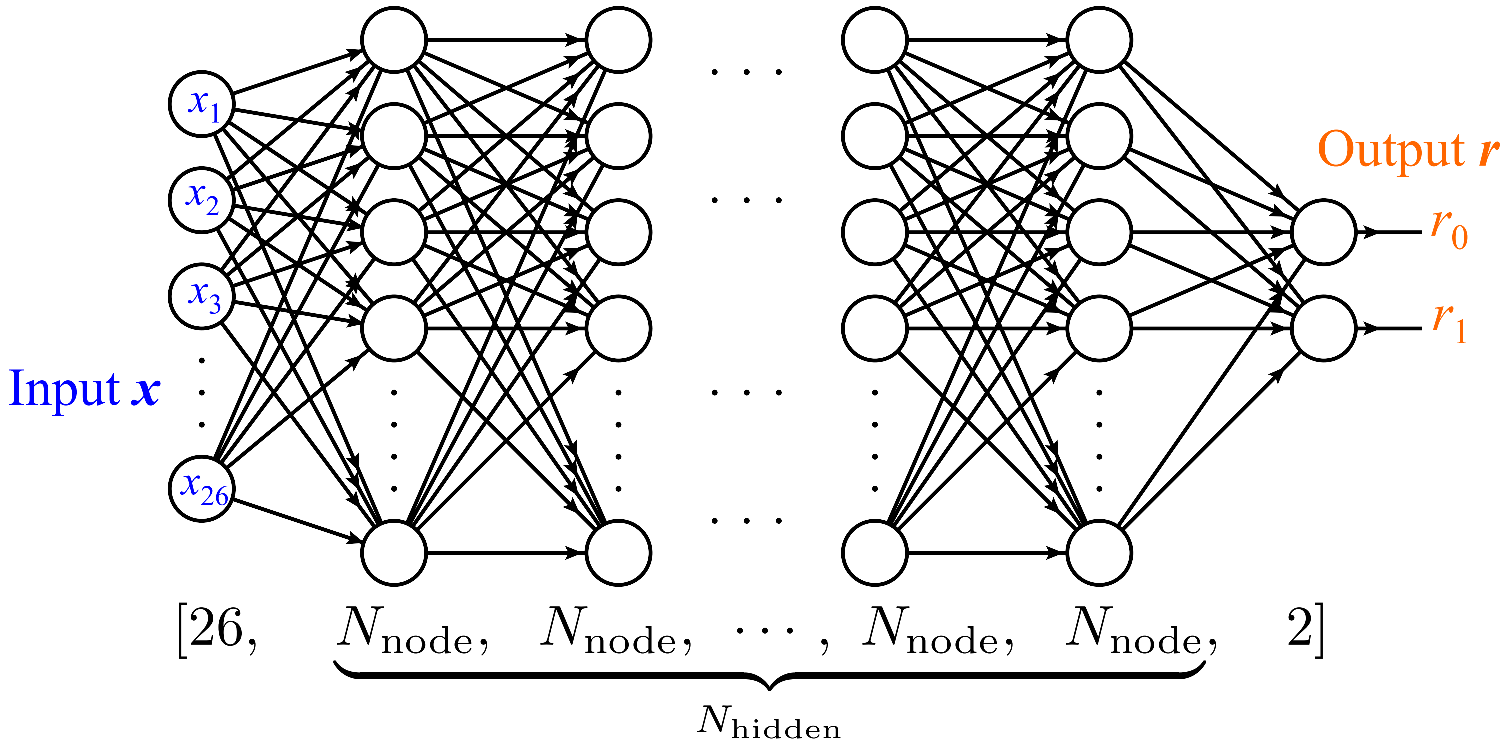


A boiological neuron



An artificial neuron

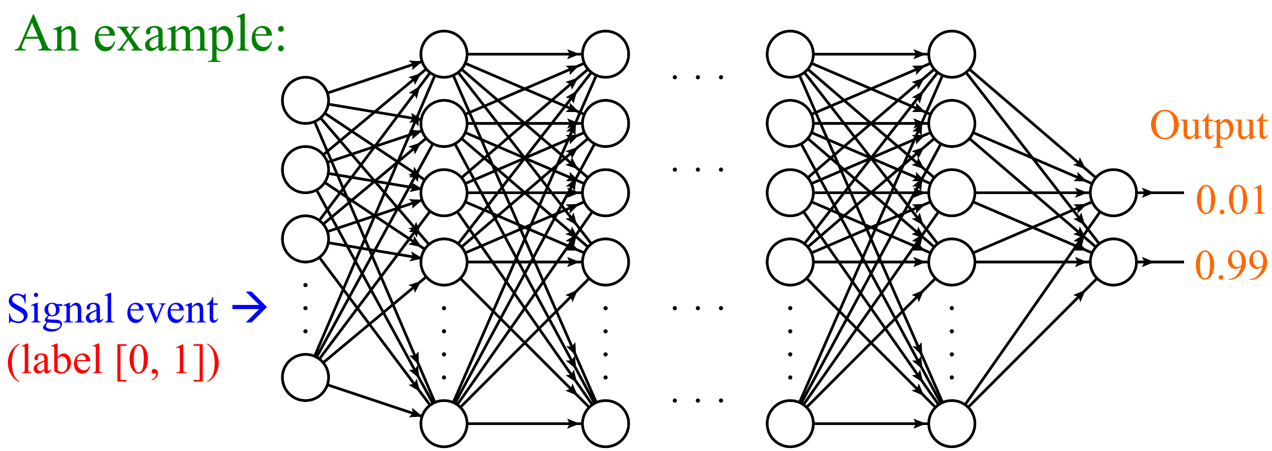
– Connecting neurons to build a neural network:



– The activation of output layer is chosen so that $r_0(\boldsymbol{x}) + r_1(\boldsymbol{x}) = 1$;

– We tried $N_{\text{node}} = 200$ or 300, $N_{\text{hidden}} = 4$ or 5, and chose the best configuration.

– Machine learning on the DNN = Based on the training dataset, use some algorithm to tune *w* [weights] and *b* [biases], such that the output ***r*** and the label ***y*** are as close as possible:



– We hope the DNN learns more information than the invariant mass M_{tt} and hence improves the efficiency at large width.

– One slide for the technical details

The configurations we tried

N_{hidden}	N_{node}	Learning rate (L_r)	Dropout rate (D_r)	Batch size (B_s)
4, 5	200, 300	0.001, 0.003	0.1, 0.2, 0.3	10^3, 10^4

Table 1: The configurations we have tried when finding the best model. For each signal model we try $2 \times 2 \times 3 \times 2 \times 2 = 48$ different configurations, and choose the one with best performance.

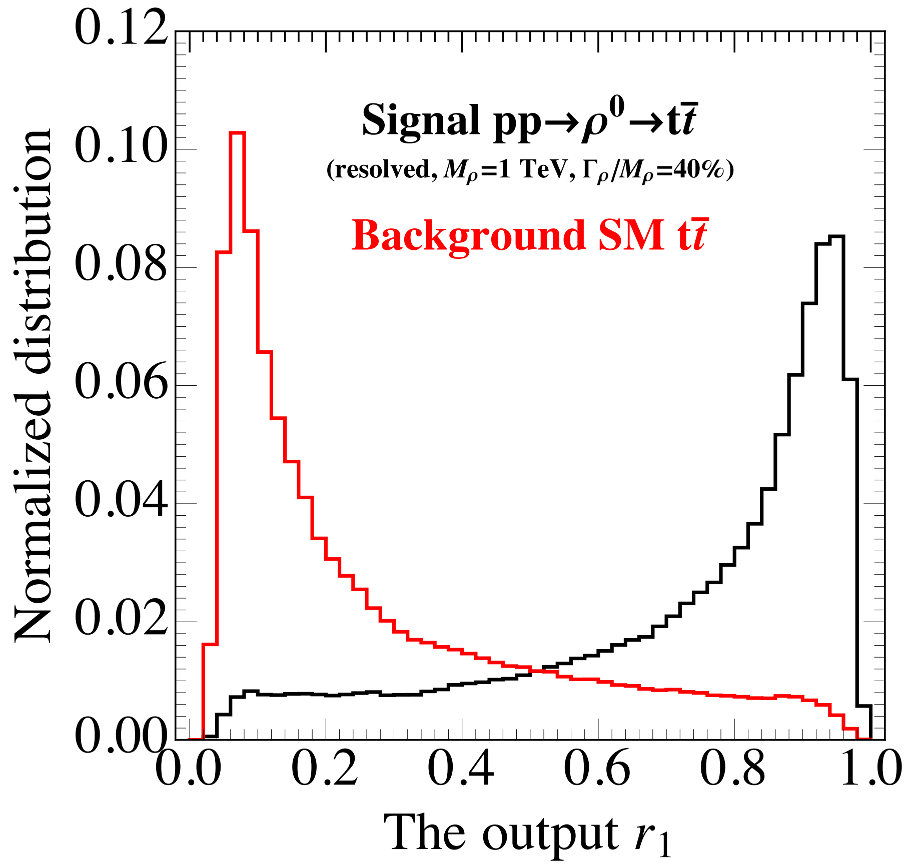
The best networks we chose

Signal model	Kinematic region	N_{hidden}, N_{node}, L_r, D_r, B_s, N_{epochs}	Accuracy reach
M1Γ1	resolved	5, 200, 0.001, 0.2, 10^3, 150	85.2%
	boosted	5, 200, 0.001, 0.2, 10^4, 55	67.9%
M1Γ2	resolved	4, 300, 0.003, 0.2, 10^3, 35	83.2%
	boosted	5, 200, 0.001, 0.2, 10^4, 45	65.8%
M1Γ3	resolved	4, 300, 0.001, 0.2, 10^3, 30	81.6%
	boosted	4, 300, 0.003, 0.2, 10^4, 30	65.1%
M1Γ4	resolved	5, 200, 0.001, 0.2, 10^3, 80	80.8%
	boosted	4, 300, 0.001, 0.2, 10^4, 20	64.3%

Table 1: The best networks for $M_\rho = 1$ TeV. N_{epochs} is the number of epochs when we cut the training; while "accuracy reach" is the accuracy for the test data.

– After training, we use another 1,000,000 events (with equal number of signal and background) to test the network.

– The well-trained network for the test dataset:



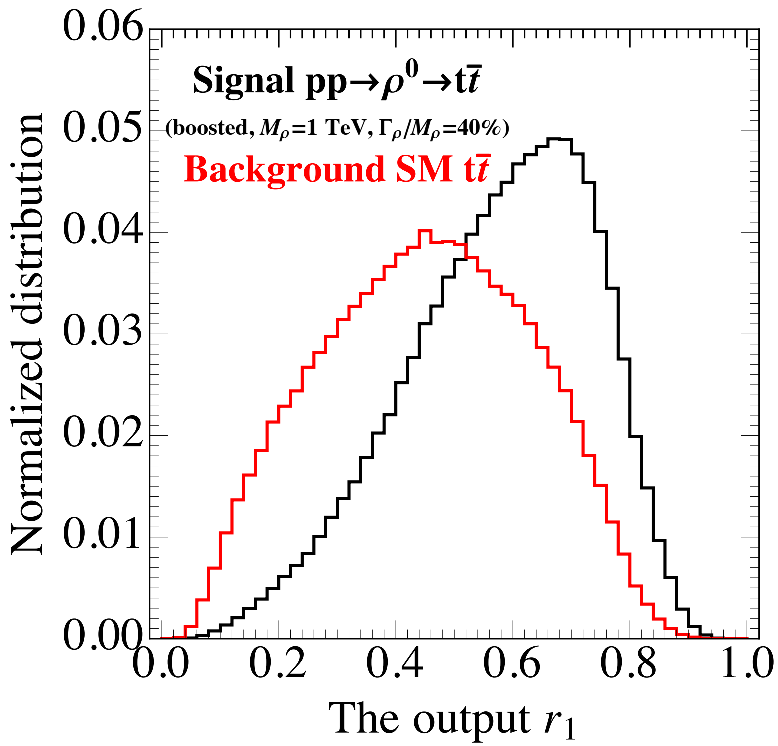
Signal and background separate very well for the test dataset!

– Training network in the boosted region: each event has 15 low-level observables:

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
E^{ℓ}	p_T^{ℓ}	η^{ℓ}	ϕ^{ℓ}	$\not{E}_T$	$\phi^{\not{E}_T}$	$E^{j_{\rm sel}}$	$p_T^{j_{\rm sel}}$	$\eta^{j_{\rm sel}}$	$\phi^{j_{\rm sel}}$	$b^{j_{\rm sel}}$	$E^{j_{\rm top}}$	$p_T^{j_{\rm top}}$	$\eta^{j_{\rm top}}$	$\phi^{j_{\rm top}}$

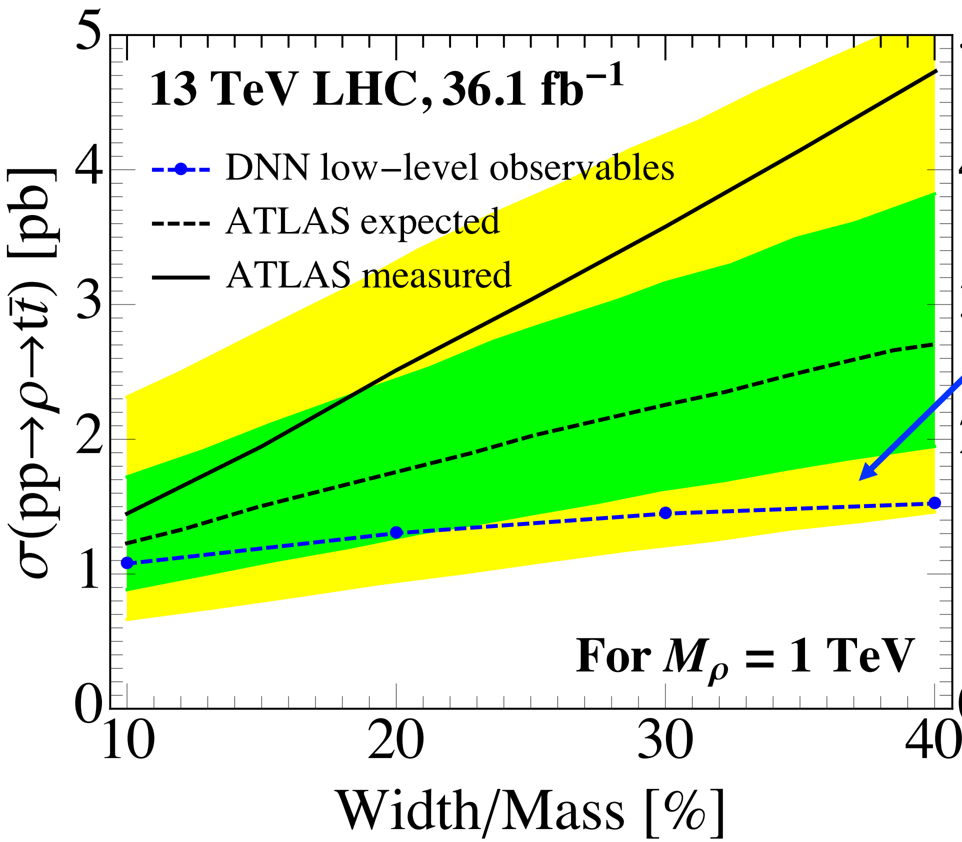
– We use a training dataset with size 800,000.

– The well-trained network for the test dataset:



Signal and background separate for the test dataset; not as good as the resolved region

- Intepret the DNN result as the cross section upper limit of the signal:
  - According to the given integrated luminosity, fit the neuron output curves and get the limits:



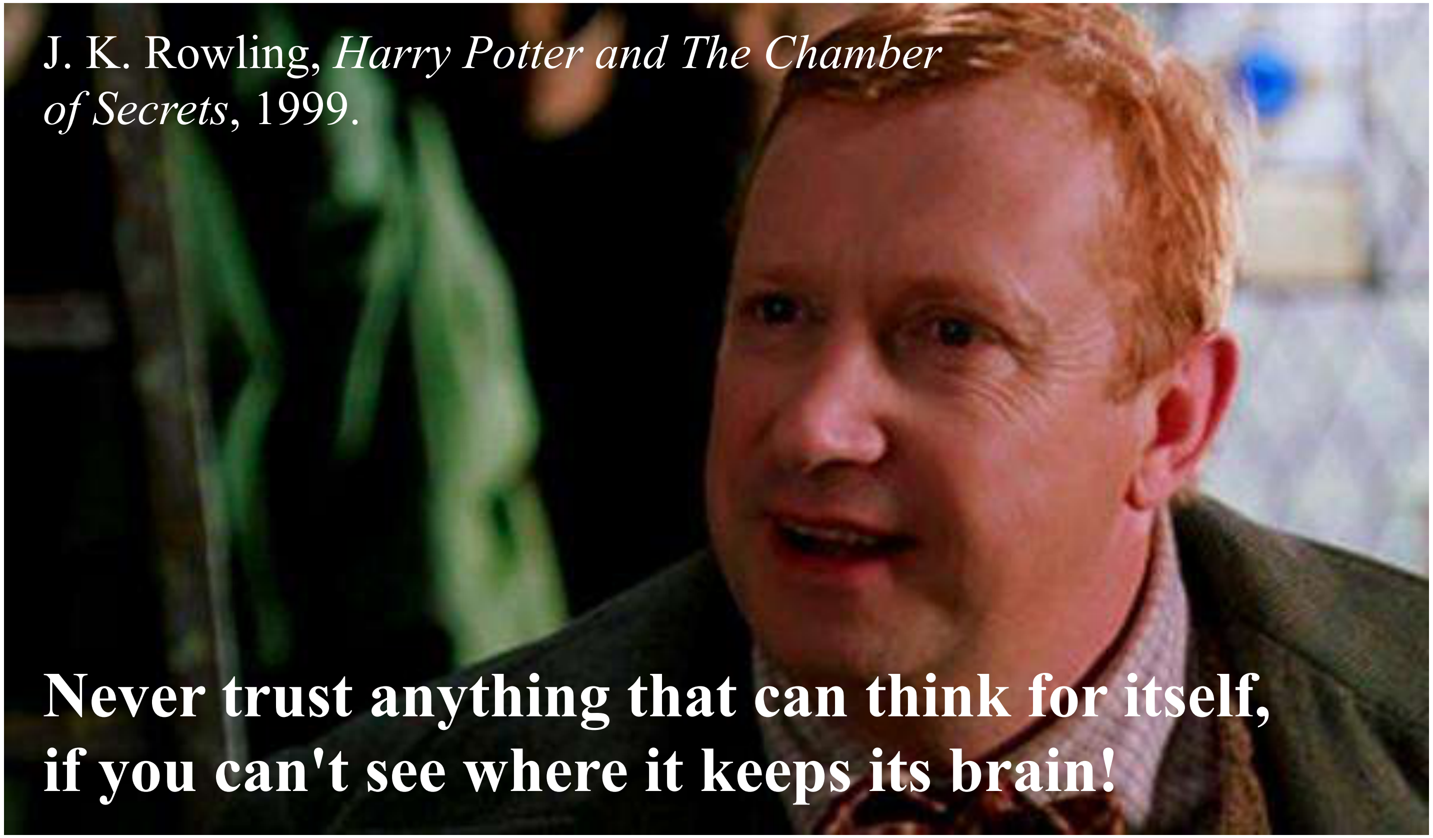
The ATLAS result: 1804.10823
The DNN result: **this work**

The DNN results are much flatter than the traditional ones!

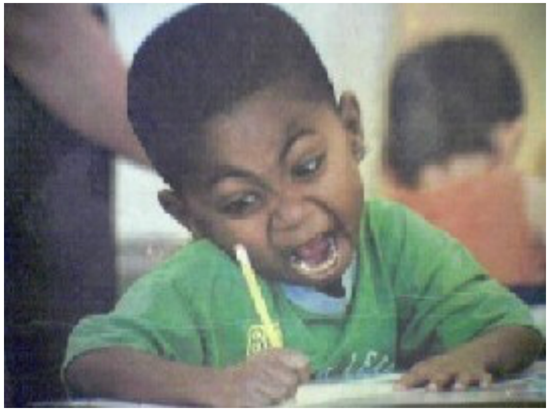
Combined resolved and boosted region; 12% systematic uncertainty for background is assumed.

- We have used the DNN to learn new physics signal from SM background, and get expected result. Is that enough?

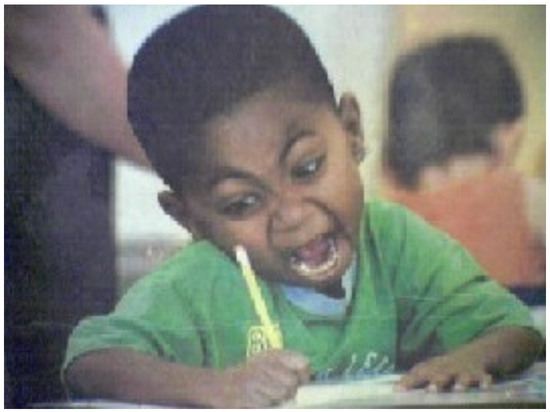
- We have used the DNN to learn new physics signal from SM background, and get expected result. Is that enough?
- NO! We have to figure out what it has learned.

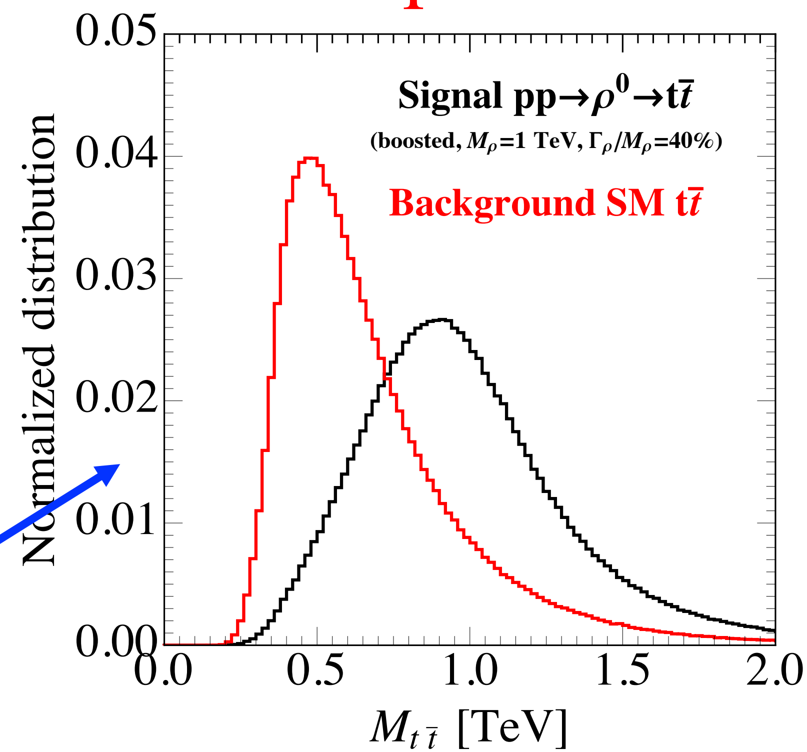


- Figuring out what the machine has learned
  - The first approach we tried: make a quiz for the machine



- Figuring out what the machine has learned
  - The first approach we tried: make a quiz for the machine



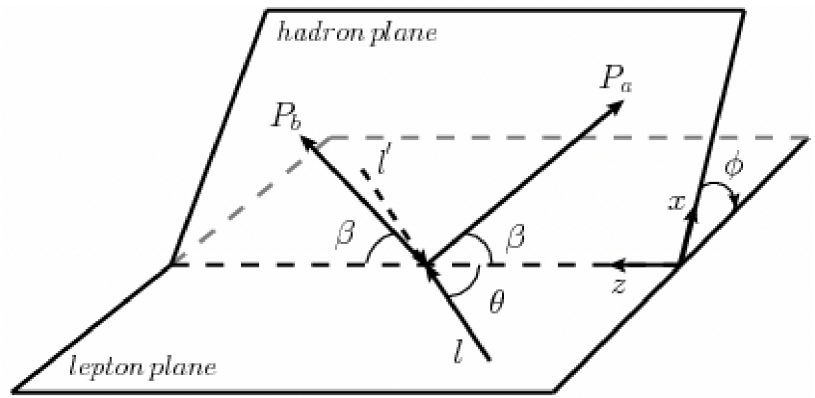


The resonance feature

  - By physical consideration, we define the 7 high-level observables:

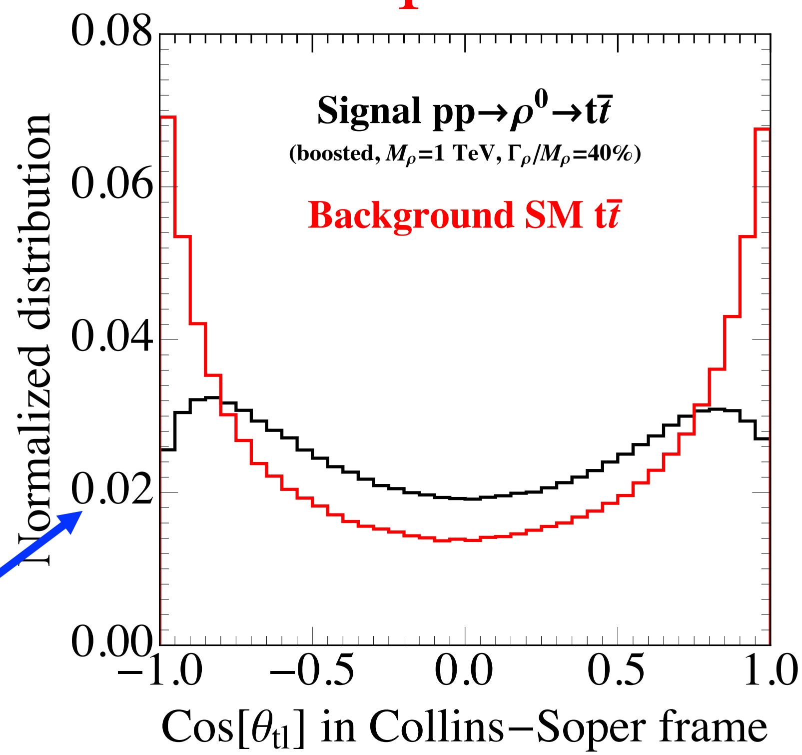
1	2	3	4	5	6	7
$M_{t\bar{t}}$	$\cos\theta_{\mathrm{tl}}^{\mathrm{CS}}$	$\cos\theta_{\mathrm{th}}^{\mathrm{CS}}$	$\phi_{\mathrm{tl}}^{\mathrm{CS}}$	$\phi_{\mathrm{th}}^{\mathrm{CS}}$	$\cos\theta_{\mathrm{tl}}^{\mathrm{Mus.}}$	$\cos\theta_{\mathrm{th}}^{\mathrm{Mus.}}$

- Figuring out what the machine has learned
  - The first approach we tried: make a quiz for the machine



* Collins-Soper frame, Phys. Rev. D16, 2219 (1977)

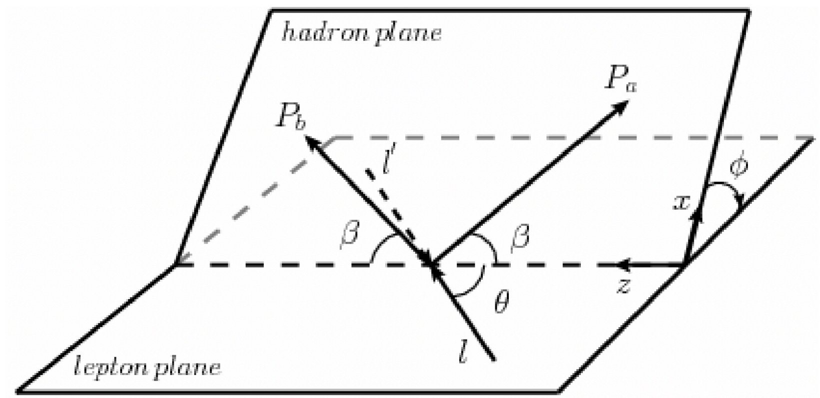
The Dell-Yan feature



  - By physical consideration, we define the 7 high-level observables:

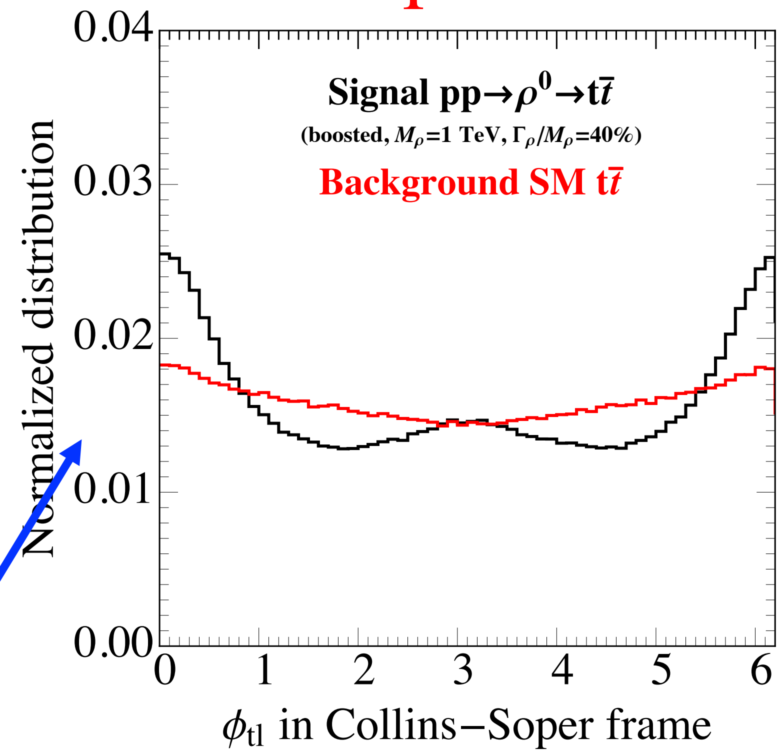
1	2	3	4	5	6	7
$M_{t\bar{t}}$	$\cos\theta_{\rm tl}^{\rm CS}$	$\cos\theta_{\rm th}^{\rm CS}$	$\phi_{\rm tl}^{\rm CS}$	$\phi_{\rm th}^{\rm CS}$	$\cos\theta_{\rm tl}^{\rm Mus.}$	$\cos\theta_{\rm th}^{\rm Mus.}$

- Figuring out what the machine has learned
  - The first approach we tried: make a quiz for the machine



* Collins-Soper frame, Phys. Rev. D16, 2219 (1977)

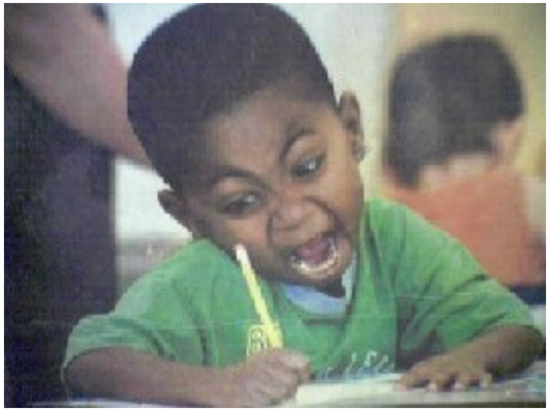
The Dell-Yan feature

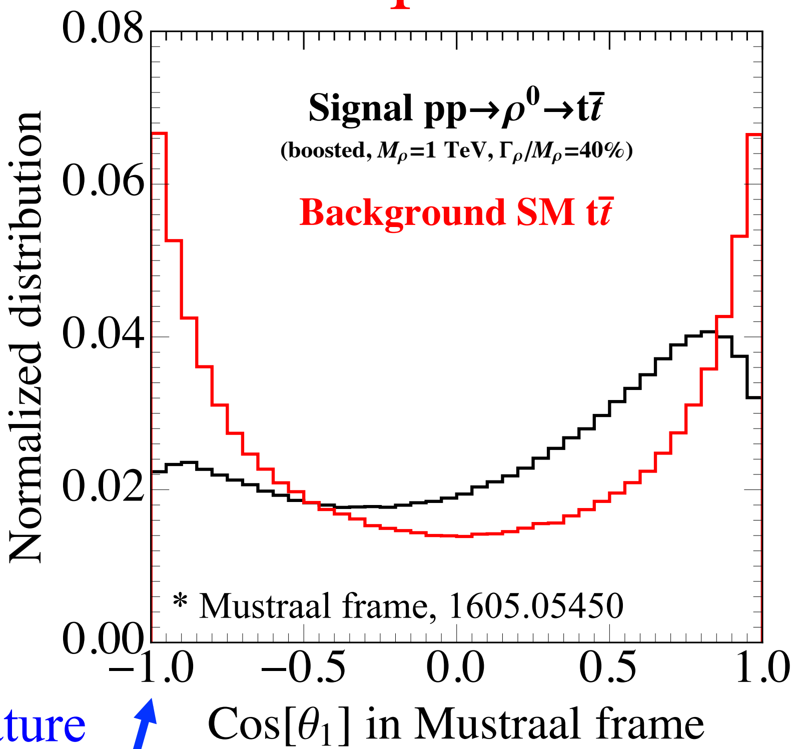


  - By physical consideration, we define the 7 high-level observables:

1	2	3	4	5	6	7
$M_{t\bar{t}}$	$\cos\theta^{\rm CS}_{\rm tl}$	$\cos\theta^{\rm CS}_{\rm th}$	$\phi^{\rm CS}_{\rm tl}$	$\phi^{\rm CS}_{\rm th}$	$\cos\theta^{\rm Mus.}_{\rm tl}$	$\cos\theta^{\rm Mus.}_{\rm th}$

- Figuring out what the machine has learned
  - The first approach we tried: make a quiz for the machine



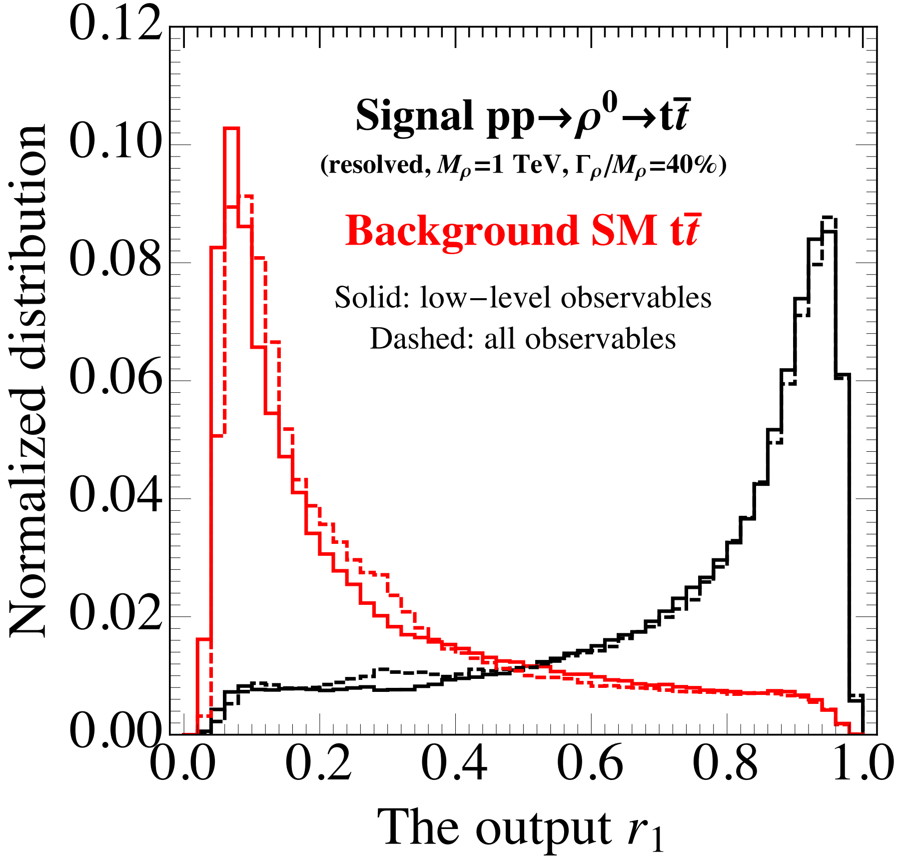


The Dell-Yan feature

  - By physical consideration, we define the 7 high-level observables:

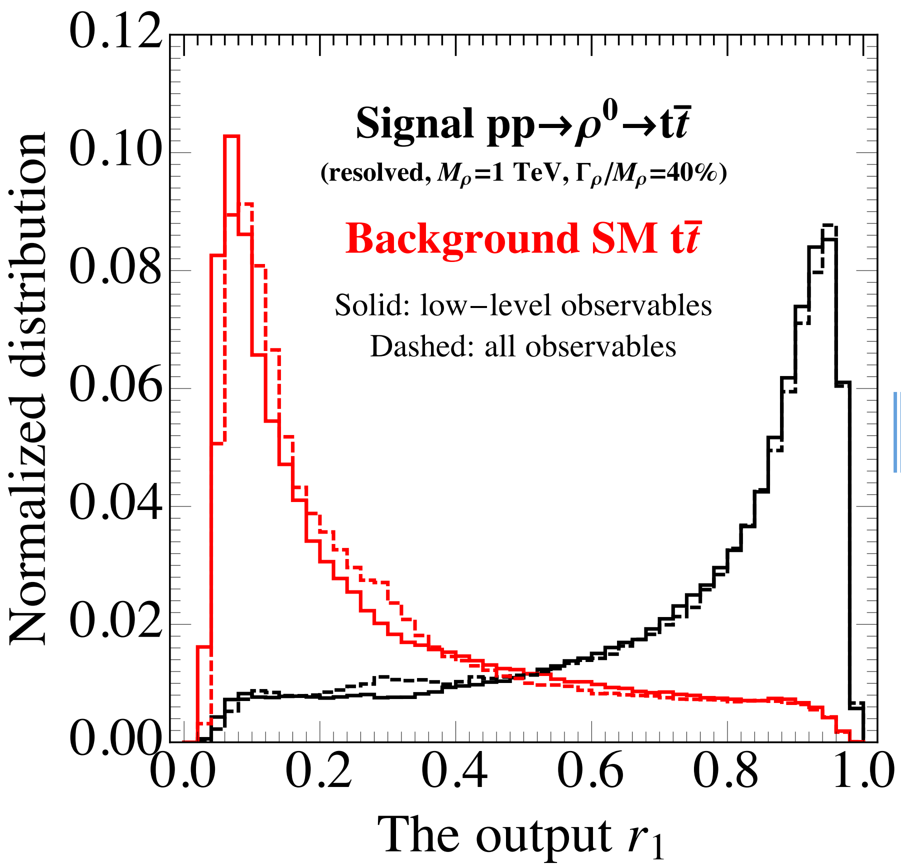
1	2	3	4	5	6	7
$M_{t\bar{t}}$	$\cos\theta_{\rm tl}^{\rm CS}$	$\cos\theta_{\rm th}^{\rm CS}$	$\phi_{\rm tl}^{\rm CS}$	$\phi_{\rm th}^{\rm CS}$	$\cos\theta_{\rm tl}^{\rm Mus.}$	$\cos\theta_{\rm th}^{\rm Mus.}$

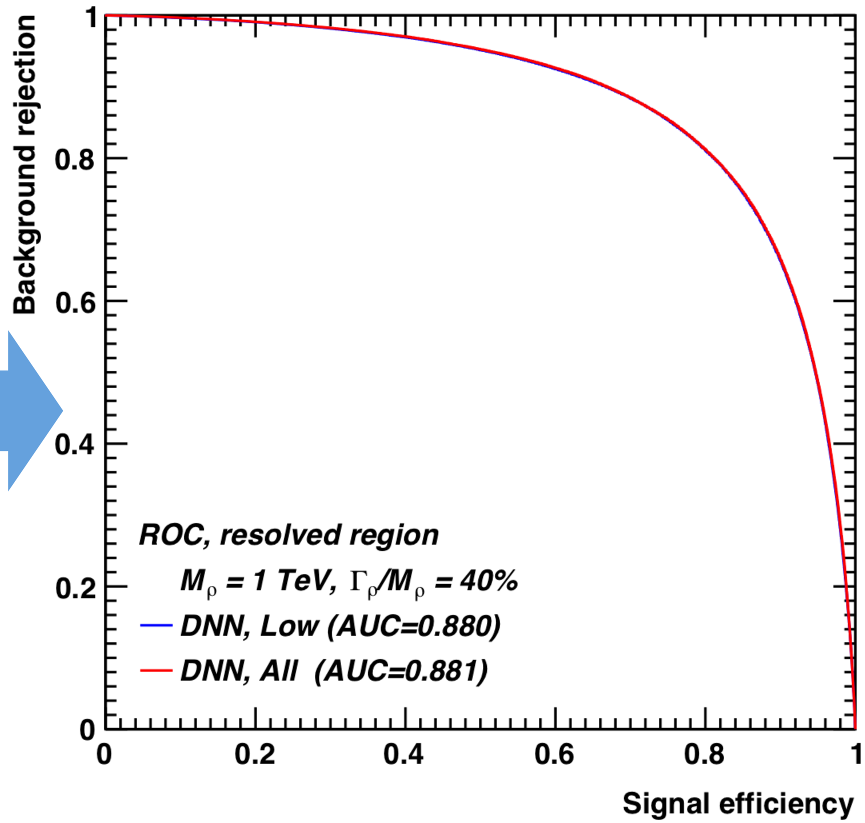
– We define all observables = low + high, and compare the training results:



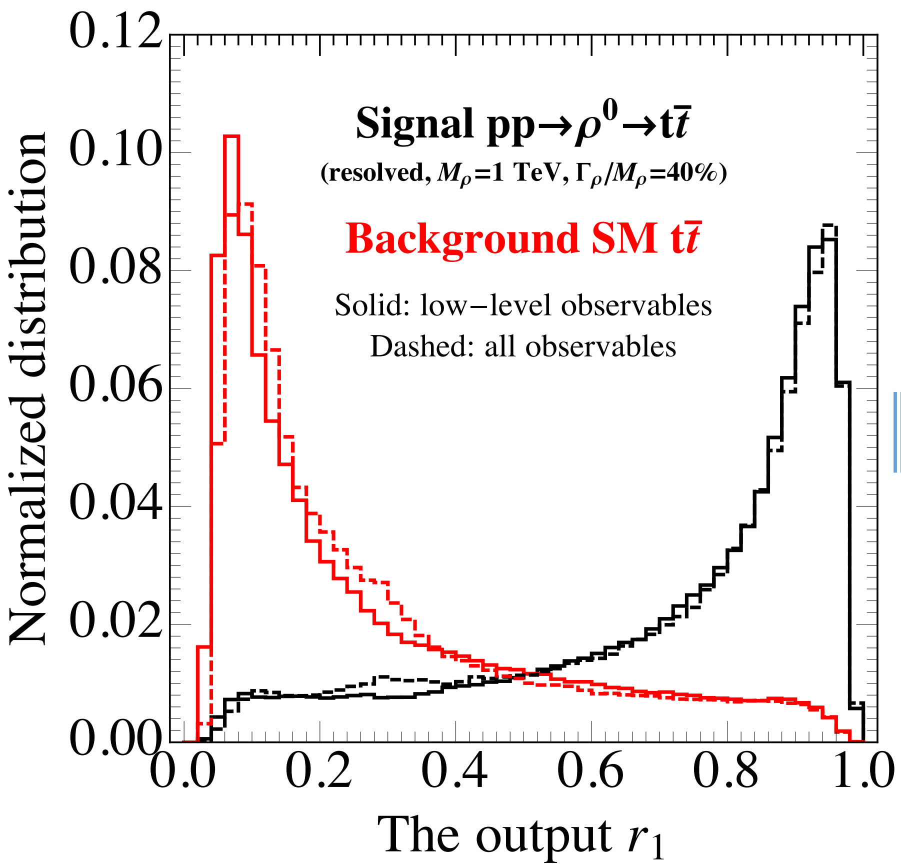
– We define all observables = low + high, and compare the training results:

ROC = receiver operating characteristic;
AUC = area under curve

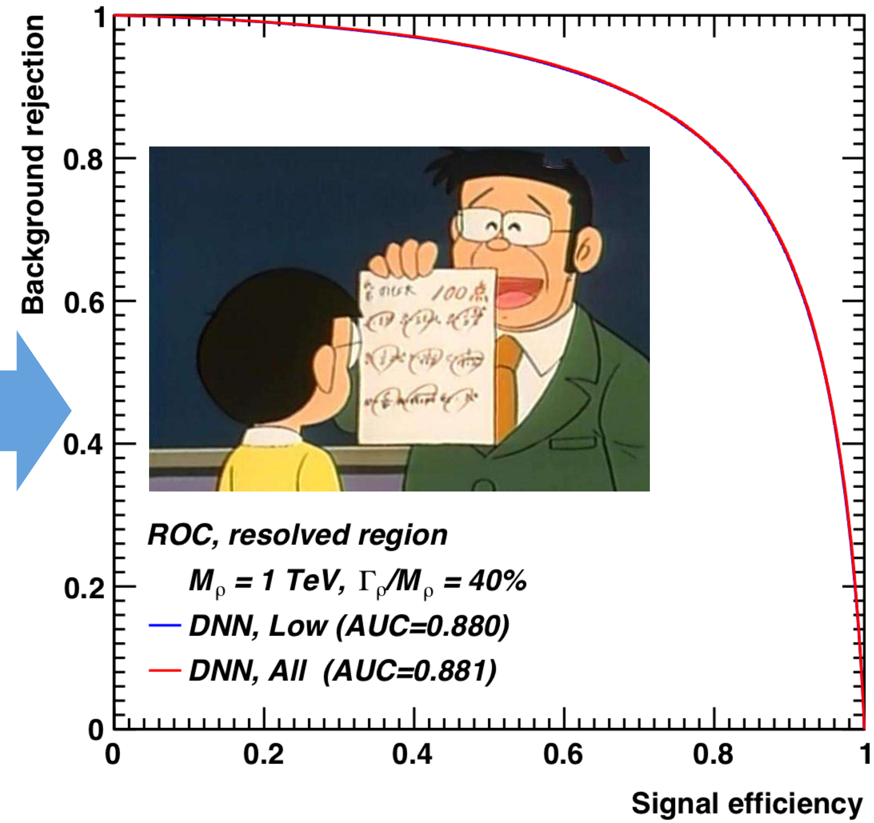




– We define all observables = low + high, and compare the training results:

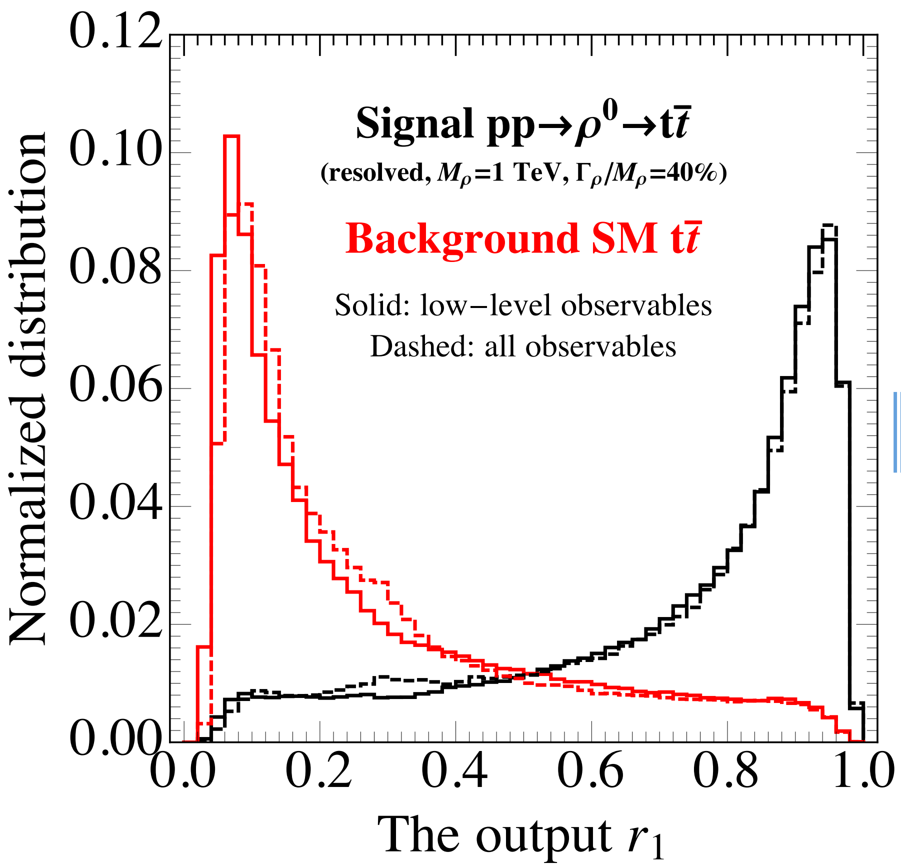


ROC = receiver operating characteristic;
AUC = area under curve

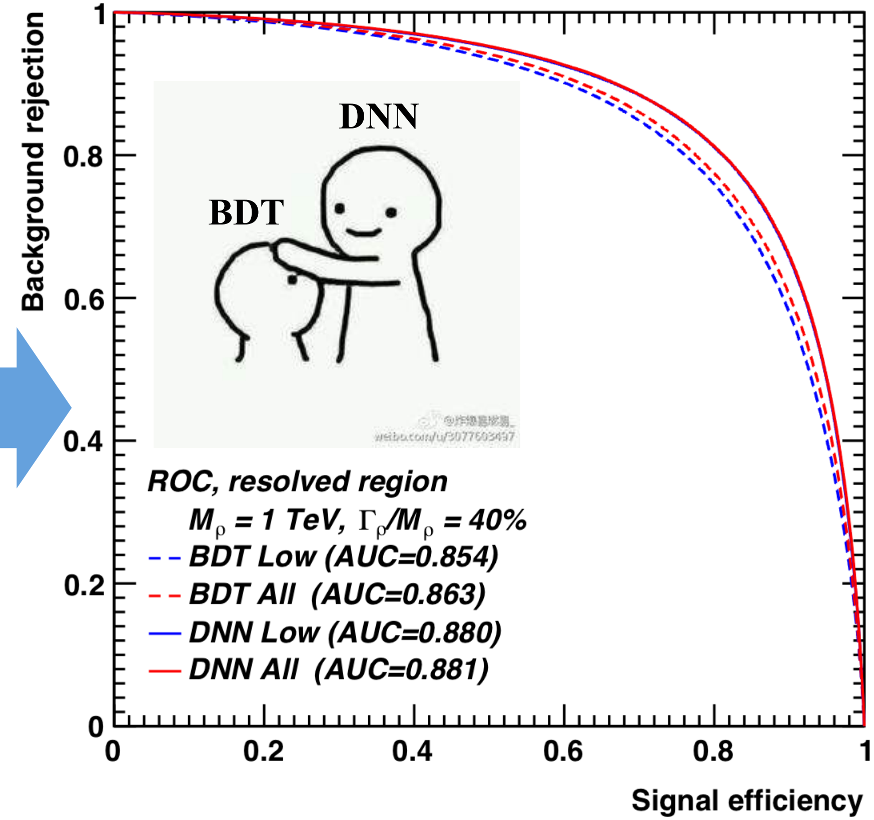


In resolved region, the machine has learned the high-level observables!

– We define all observables = low + high, and compare the training results:

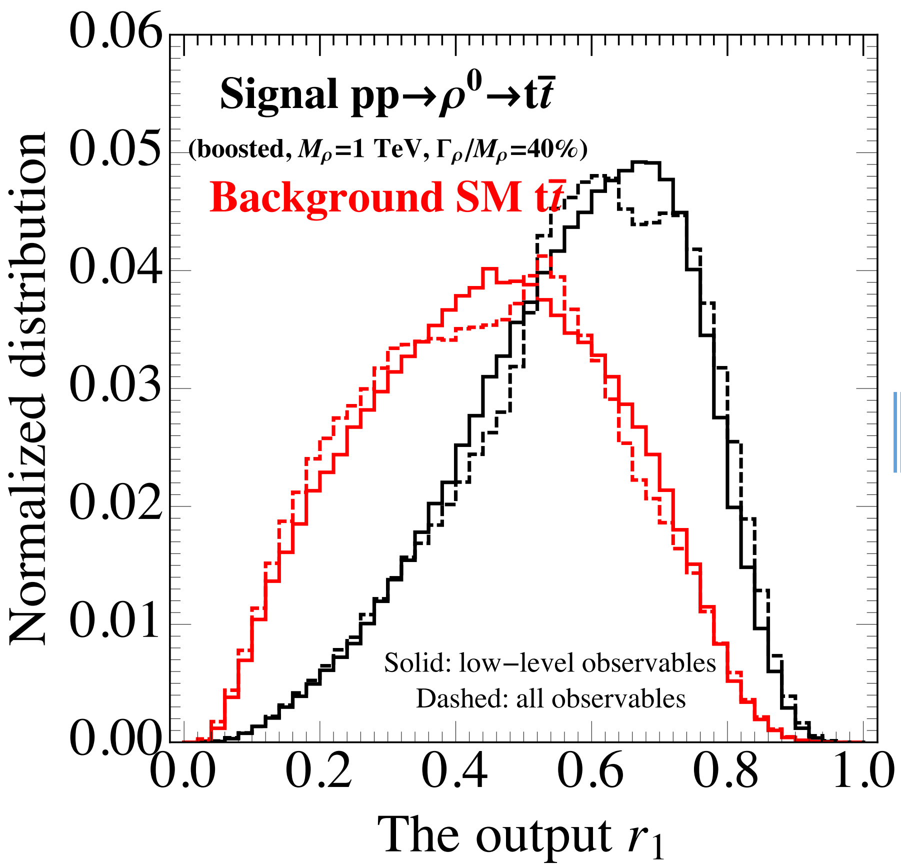


ROC = receiver operating characteristic;
AUC = area under curve

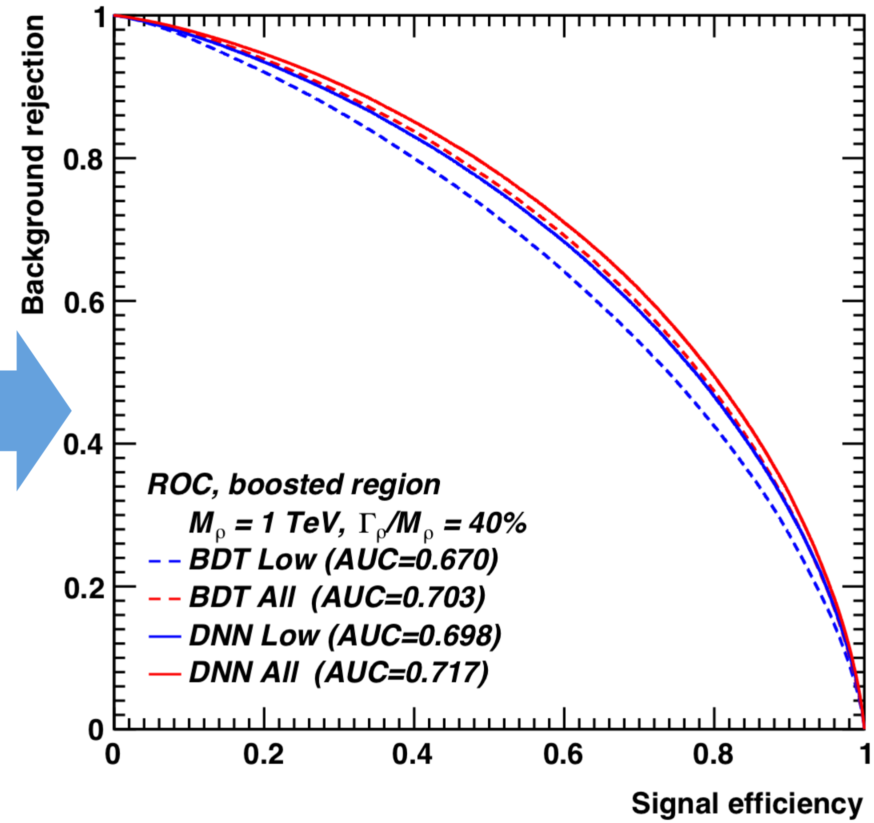


The BDT is not able to learn the high-level observables; and it is worse than DNN.

– We define all observables = low + high, and compare the training results:



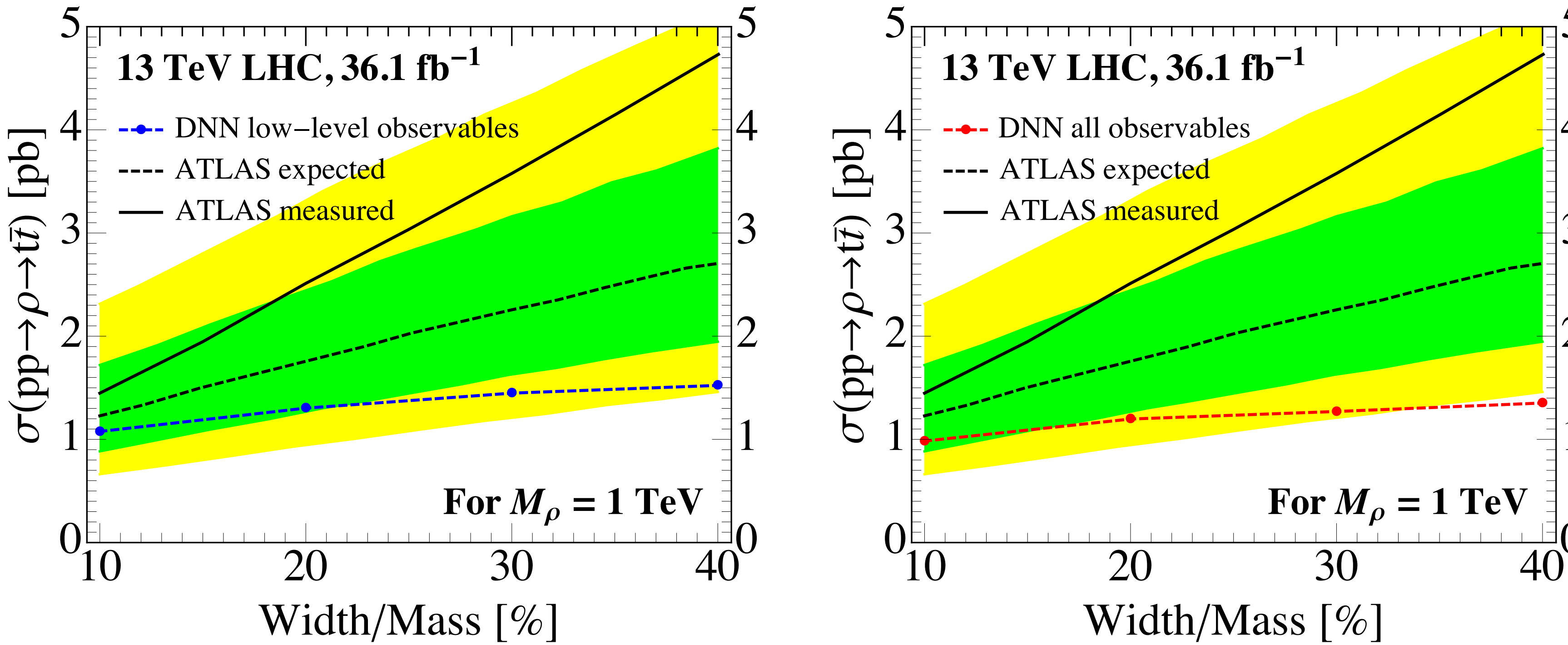
ROC = receiver operating characteristic;
AUC = area under curve



In boosted region, the DNN can learn just part of the high-level observables; it is still better than BDT.

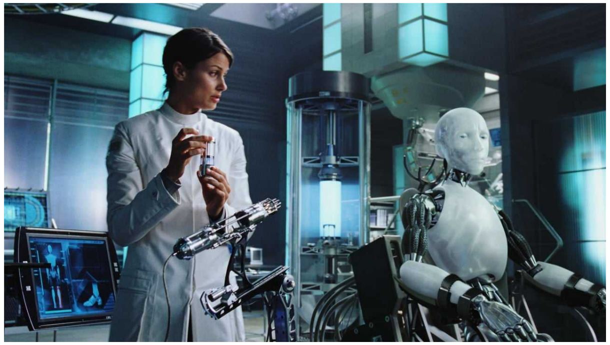
– compare the upper limits:

The ATLAS result: 1804.10823
The DNN results: **this work**

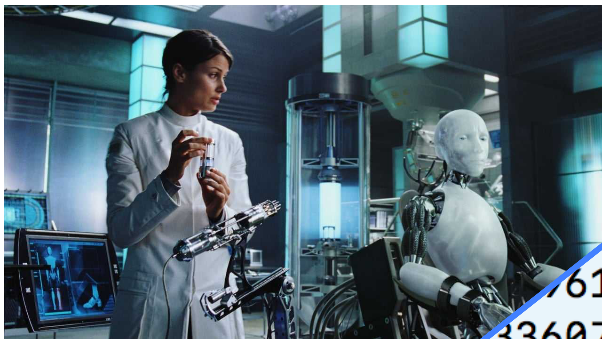


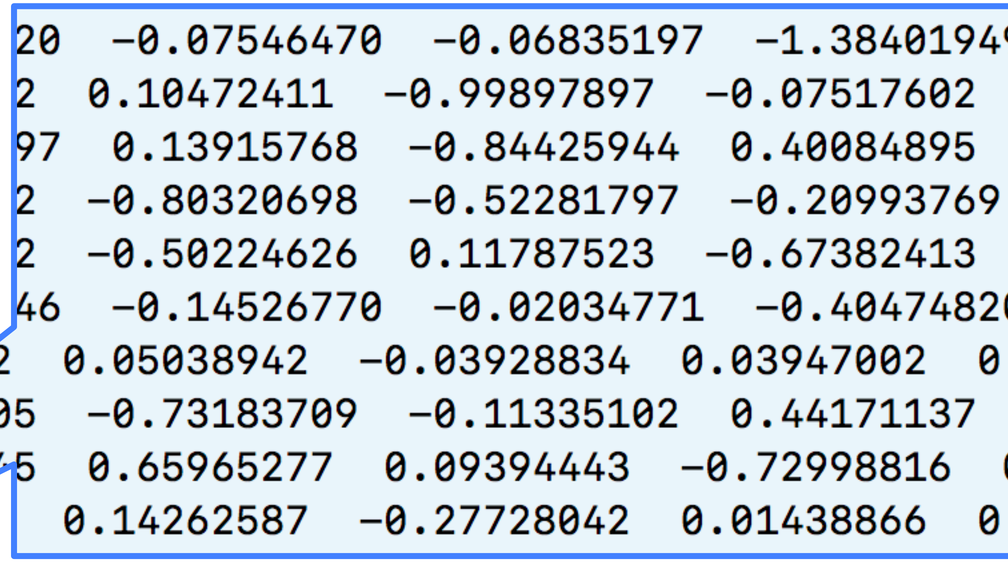
– The DNN result is slightly improved when the high-level observabels are included.

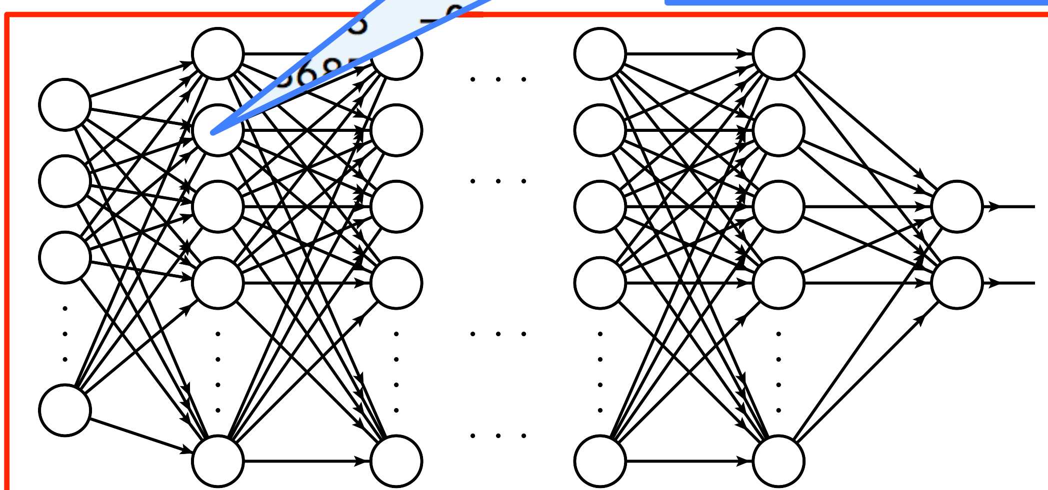
- Figuring out what the machine has learned
  - The second approach we tried: disassemble the machine



- Figuring out what the machine has learned
  - The second approach we tried: disassemble the machine

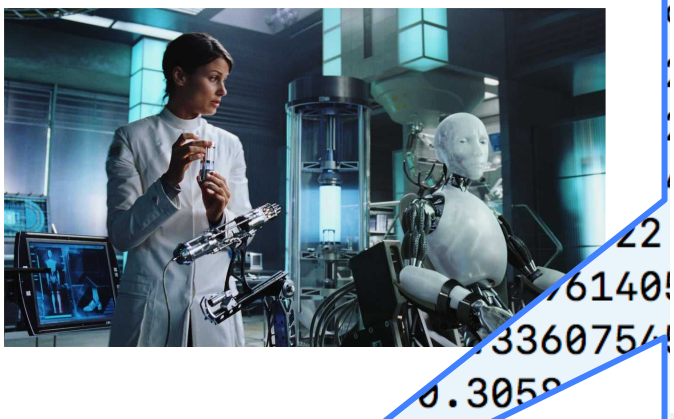






The weights and biases of the DNN

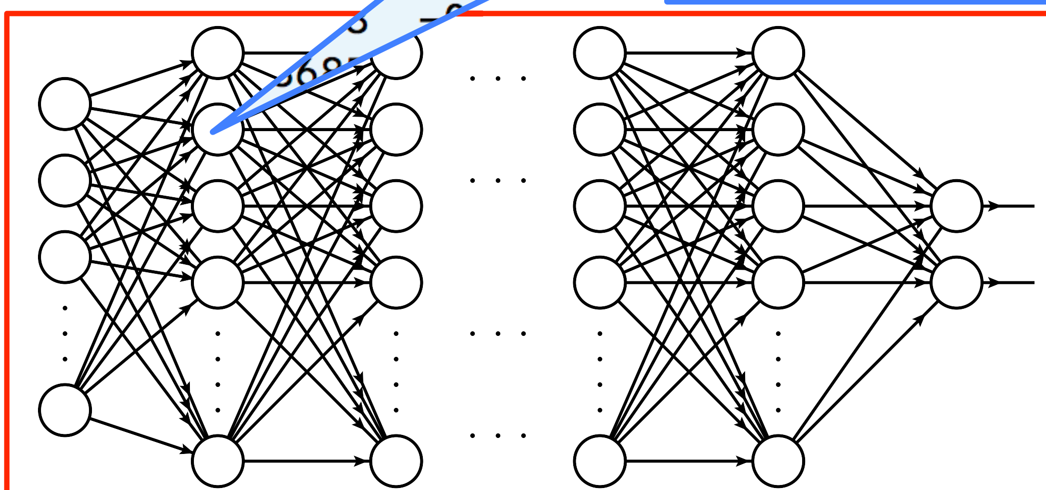
- Figuring out what the machine has learned
  - The second approach we tried: disassemble the machine



If the fist hidden layer has 200 neurons, the 1st weights $w_{ij}^{(1)}$ form a 26×200 matrix. Define

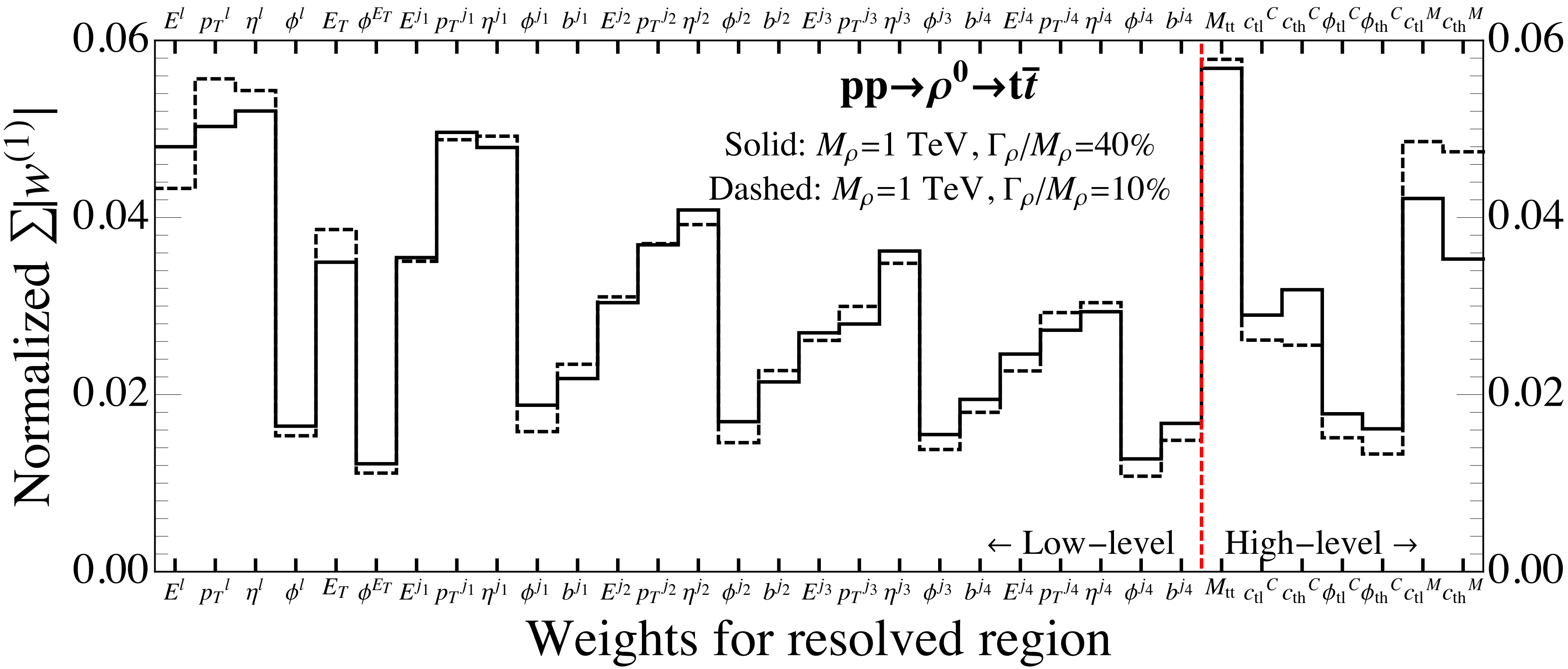
$$w_i = \frac{\sum_{j=1}^{200} |w_{ij}^{(1)}|}{\sum_{i=1,j=1}^{26,200} |w_{ij}^{(1)}|}$$

Then for each input observable we have a weight.



The weights and biases of the DNN

– Disassembling the machine



Low-level

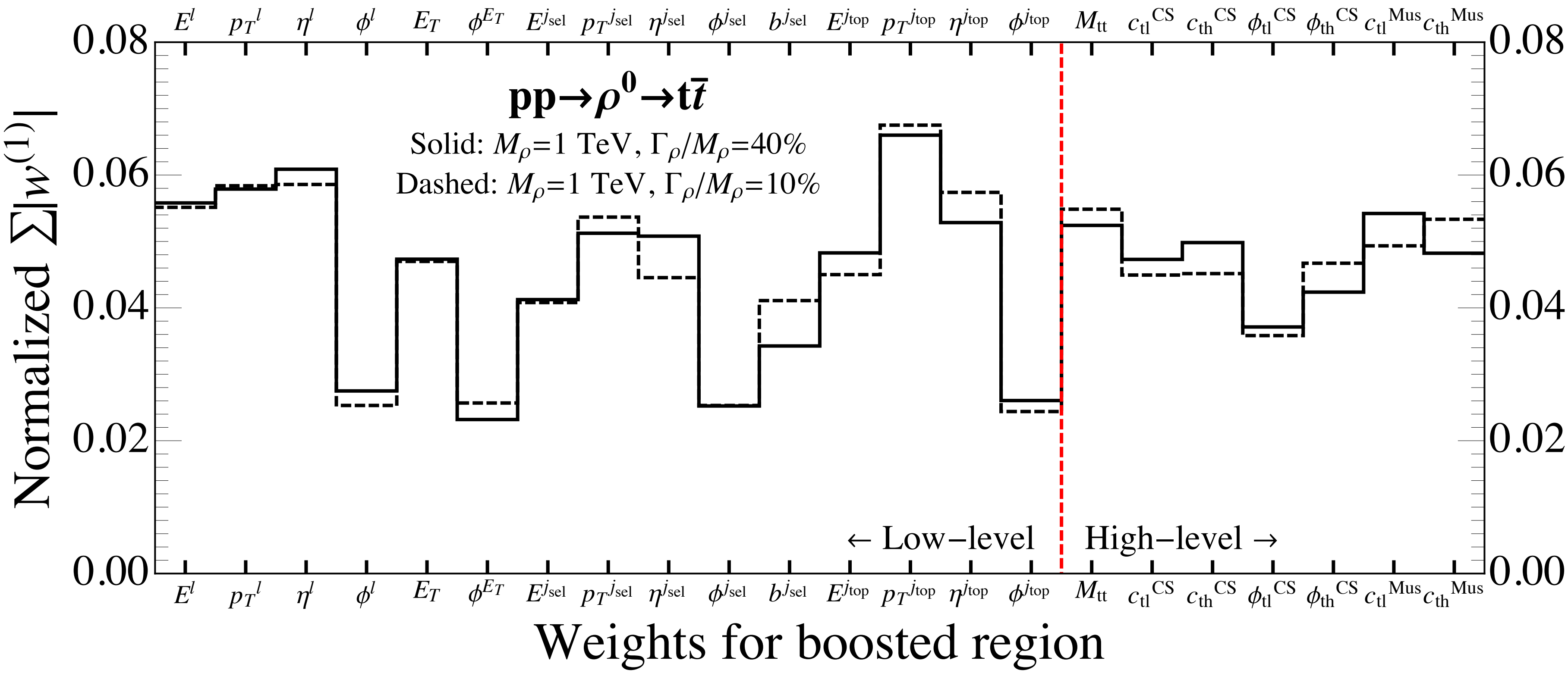
1	2	3	4	5	6	7	8	9	10	11	12	13
E^ℓ	p_T^ℓ	η^ℓ	ϕ^ℓ	$\cancel{E}_T$	$\phi^{\cancel{E}_T}$	E^{j_1}	$p_T^{j_1}$	η^{j_1}	ϕ^{j_1}	b^{j_1}	E^{j_2}	$p_T^{j_2}$

14	15	16	17	18	19	20	21	22	23	24	25	26
η^{j_2}	ϕ^{j_2}	b^{j_2}	E^{j_3}	$p_T^{j_3}$	η^{j_3}	ϕ^{j_3}	b^{j_3}	E^{j_4}	$p_T^{j_4}$	η^{j_4}	ϕ^{j_4}	b^{j_4}

High-level

1	2	3	4	5	6	7
$M_{t\bar{t}}$	$\cos\theta_{\rm tl}^{\rm CS}$	$\cos\theta_{\rm th}^{\rm CS}$	$\phi_{\rm tl}^{\rm CS}$	$\phi_{\rm th}^{\rm CS}$	$\cos\theta_{\rm tl}^{\rm Mus.}$	$\cos\theta_{\rm th}^{\rm Mus.}$

– Disassembling the machine



Low-level

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
E^ℓ	p_T^ℓ	η^ℓ	ϕ^ℓ	$\not{E}_T$	$\phi^{\not{E}_T}$	E^{j_sel}	$p_T^{j_\text{sel}}$	η^{j_sel}	ϕ^{j_sel}	b^{j_sel}	E^{j_top}	$p_T^{j_\text{top}}$	η^{j_top}	ϕ^{j_top}

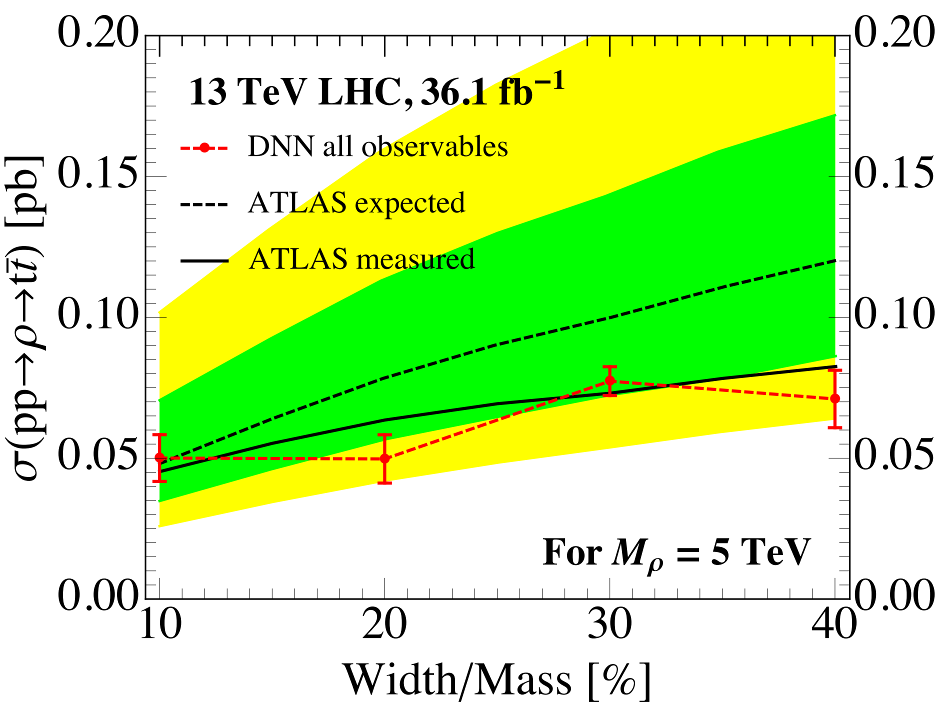
High-level

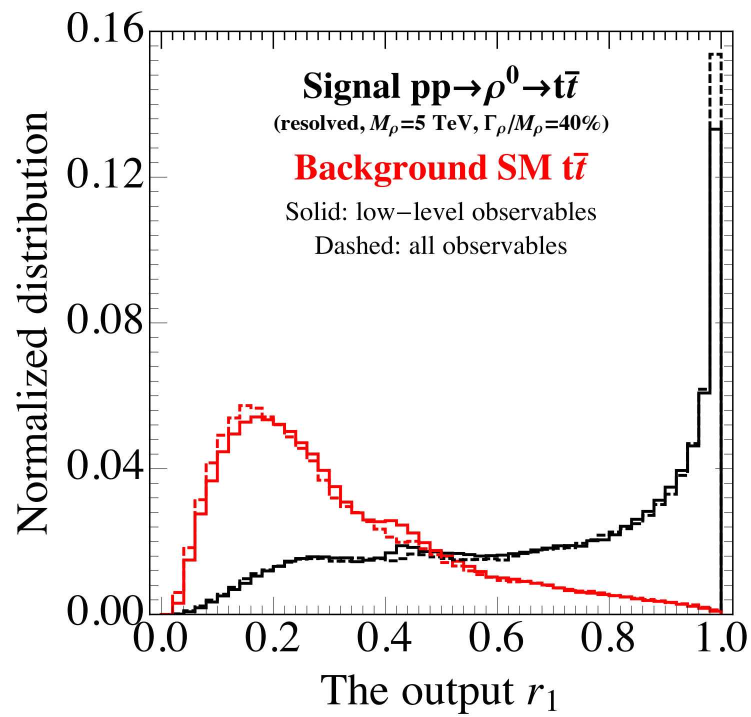
1	2	3	4	5	6	7
$M_{t\bar{t}}$	$\cos\theta_\text{tl}^\text{CS}$	$\cos\theta_\text{th}^\text{CS}$	ϕ_tl^CS	ϕ_th^CS	$\cos\theta_\text{tl}^\text{Mus.}$	$\cos\theta_\text{th}^\text{Mus.}$

➢One slide for the 5 TeV case

The ATLAS result: 1804.10823

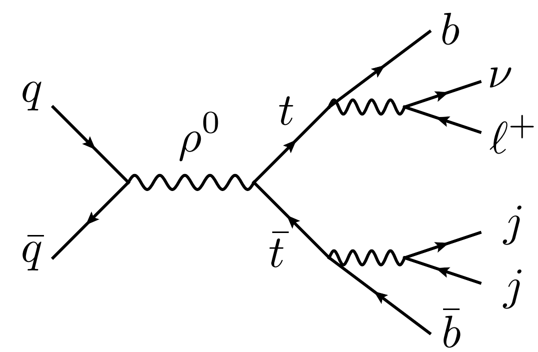
The DNN results: **this work**





➢ Conclusion

1. DNN makes use of all observables of the final state and reach a better cross section upper limit in the *ttbar* resonance searching;



2. In resolved region, DNN can learn all high-level observables via the low-level observables, while in boosted region it can learn part of the high-level observables;
3. In any case DNN works better than BDT.

Thank you!