Poster of HPNP 2019, February 21, Osaka, Japan

In collaboration with Sunghoon Jung and Dongsub Lee

Deep learning in searching for a broad $t\bar{t}$ resonance at the LHC

Ke-Pan Xie [kpxie@snu.ac.kr]（谢柯盼）
Seoul National University

1. We use deep neuron network (DNN) to search for a broad $t\bar{t}$ resonance.
2. The DNN makes use of the kinematic information of all reconstructed objects in the final state, thus achieves a better bound than the traditional approach.
3. We try two approaches to test what the DNN has learned.

1. The broad $t\bar{t}$ resonance

- Exists generally in strongly interacting New Physics models with top-quark portal;
- An example (our **benchmark**):

$$\mathcal{L} = -\frac{1}{4}\rho_{\mu\nu}\rho^{\mu\nu} + \frac{m_\rho^2}{2g_\rho^2}(g_\rho\rho_\mu - g_1 B_\mu)^2 + \bar{t}_R\gamma^\mu t_R(g_\rho\rho_\mu - g_1 B_\mu),$$

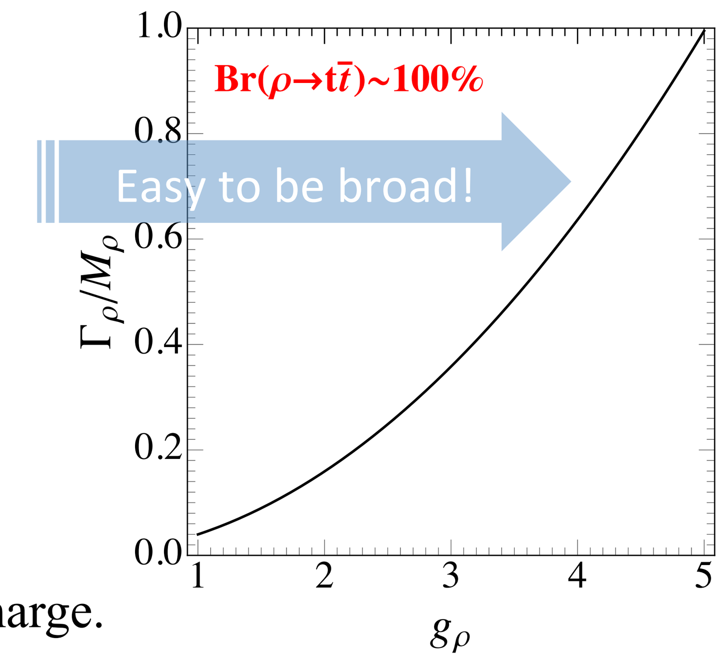
- The (gauge singlet) spin-1 resonance ρ:

$$M_\rho = m_\rho + \mathcal{O}(v);$$

$$\text{Vertex}(\rho t_R \bar{t}_R) \sim g_\rho;$$ Top portal

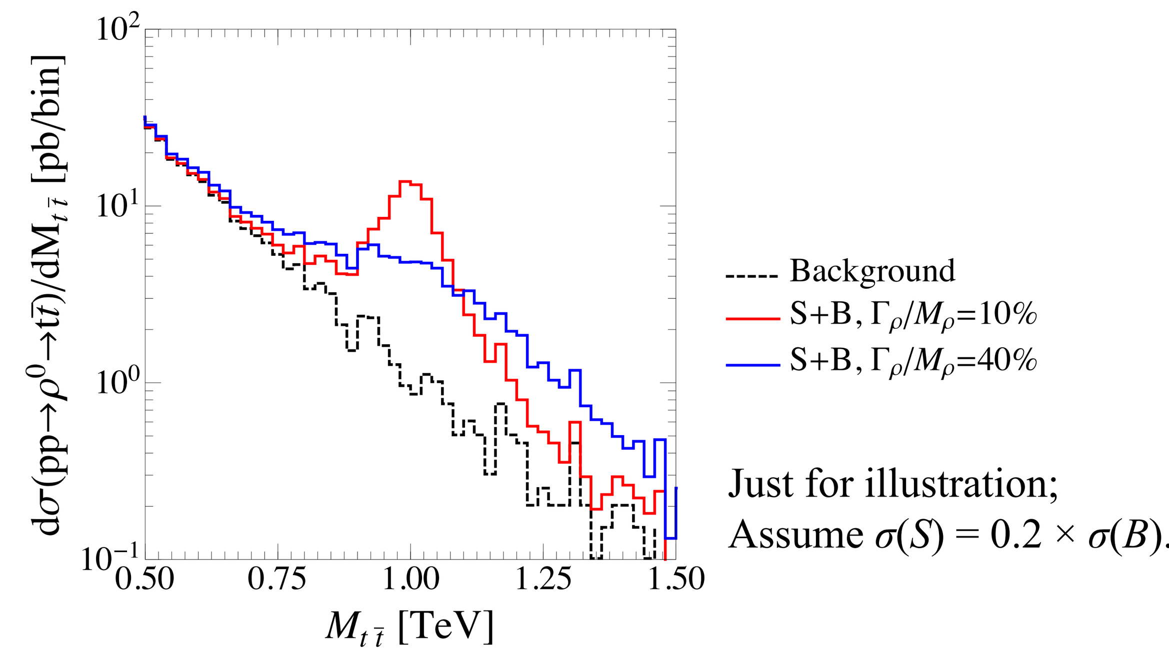
$$\text{Vertex}(\rho f\bar{f}) \sim Y_f\frac{g'^2}{g_\rho};$$ Through ρ-B mixing

f denotes the SM fermions and Y_f is the hypercharge.

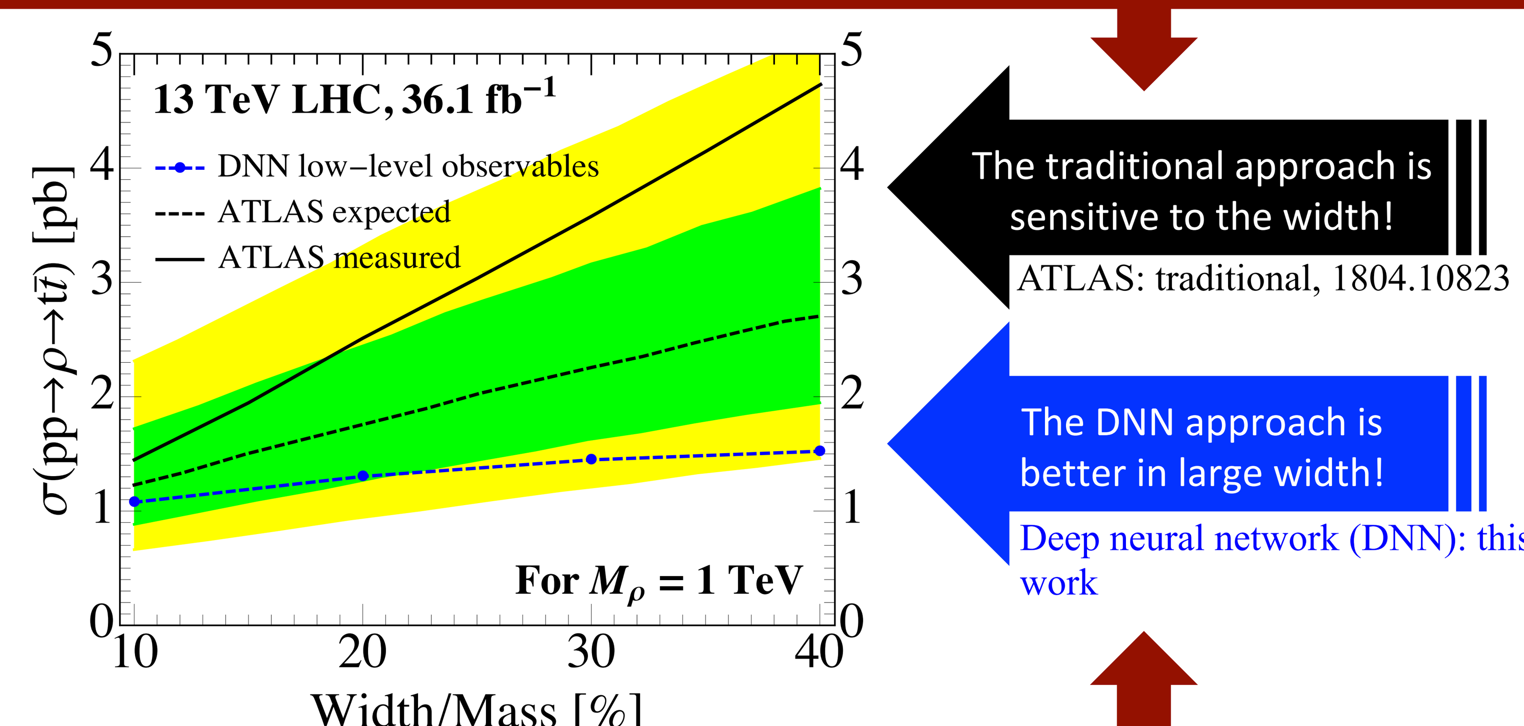


2. Searching for a $t\bar{t}$ resonance: traditional approach

- To fit the invariant mass distribution of the $t\bar{t}$ system:



- In the traditional approach, only one observable $M_{t\bar{t}}$ is used.
- As a result, the measured bound is **worse** at large width region, because the resonant peak is smeared out.

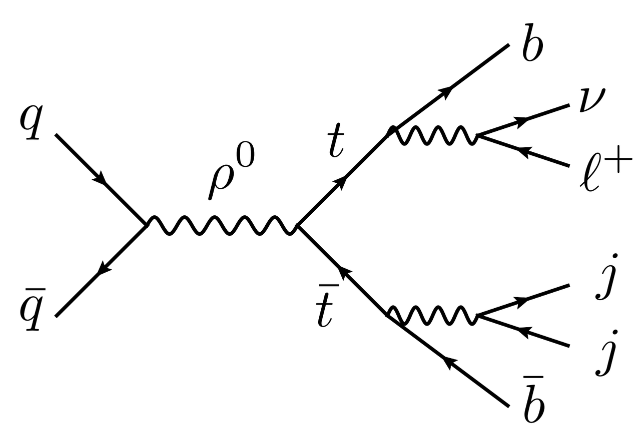


3. Searching for a $t\bar{t}$ resonance: deep learning approach

- The process under consideration: Signal : $pp \to \rho^0 \to t\bar{t} \to 1\ell^\pm + \text{jets}$
  Background : SM $pp \to t\bar{t} \to 1\ell^\pm + \text{jets}$
- Parameter benchmarks:

$$M_\rho = 1,\ 5\ \text{TeV};\quad \Gamma_\rho/M_\rho = 10\%,\ 20\%,\ 30\%,\ 40\%,$$

- Two kinematic regions:



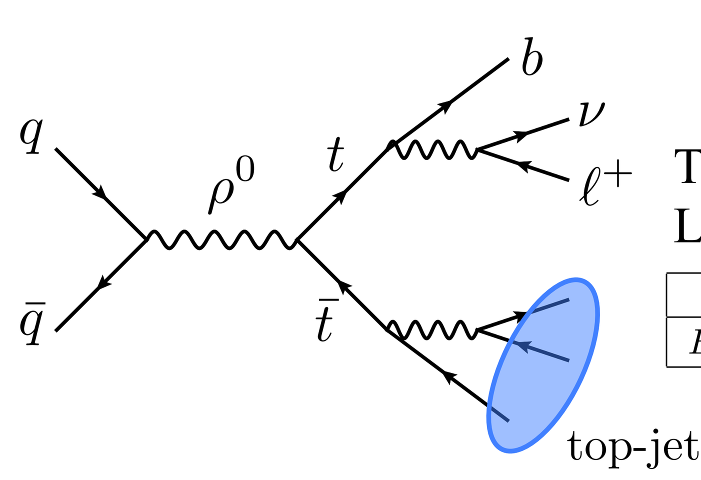
The **resolved** region: for 1 TeV resonance;
Low-level features for training

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}

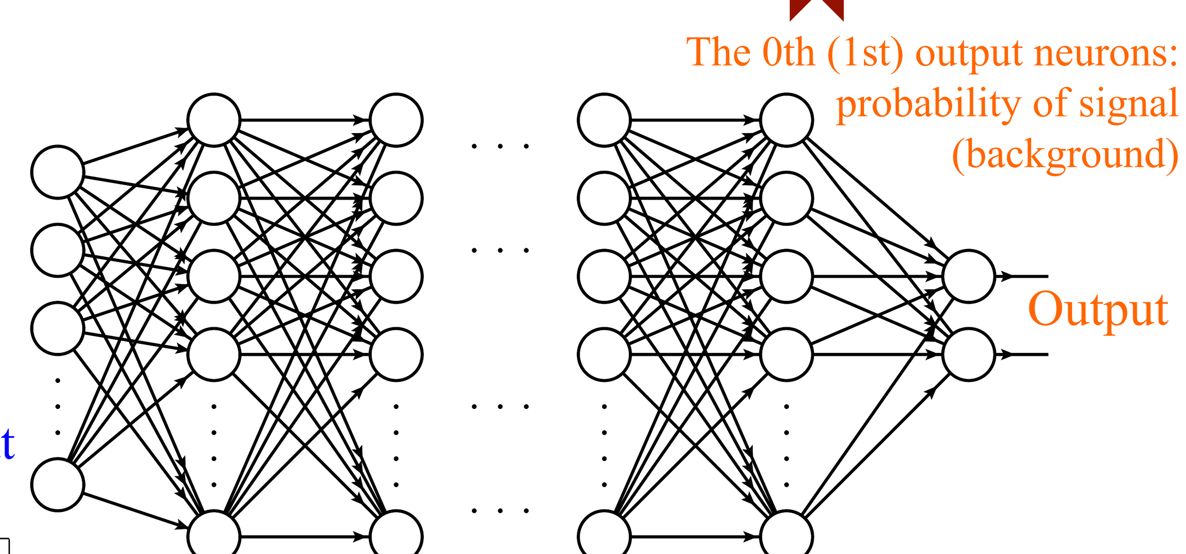
(b^j: 1 for a b-tagged jet while 0 for an un-tag jet.)

Training on fully-connected neural network



The **boosted** region: for 1 and 5 TeV resonance;
Low-level features for training

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



For deep learning: we tried 4 or 5 hidden layers, with 200 or 300 neurons per layer.

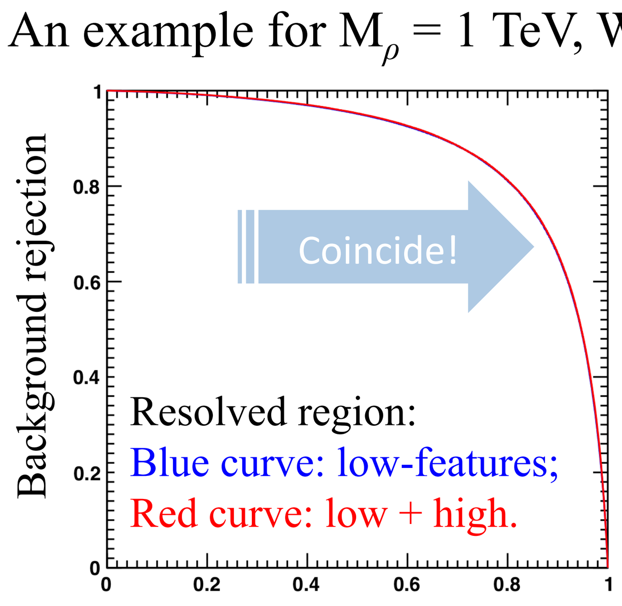
4. Figuring out what the machine has learned

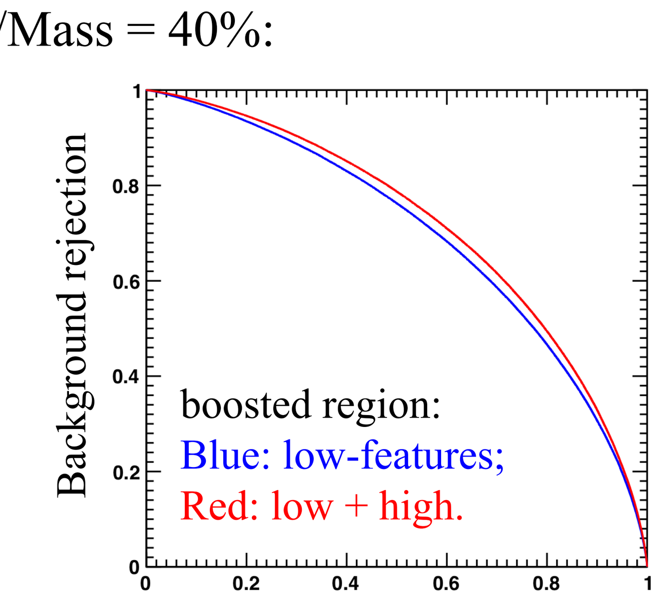
➢ The **first** approach: use the receiver operating characteristic (ROC) curves to test whether it has learned some specific physical observables.

- We define 7 high-level (i.e. expert-defined, well-motivated) observables to test: invariant mass of top-pair; angles in Collins-Soper frame (see Phys. Rev. D16, 2219 (1977)); angles in Mustraal frame (see 1605.05450).

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 found that the neural network can learn all high-level observables via the low-level features in the resolved region; while for the boosted region, it can only learn part of the high-level features due to the tight cut.
- An example for M_ρ = 1 TeV, Width/Mass = 40%:





- If the red and blue curves coincide, that means the network has learned all high-level observables from the low-level ones.

➢ The **second** approach: disassemble the network!

- We found that the 1st hidden layer typically has a learning speed several times larger than other layers. Motivated by this, we use the weights from the 1st hidden layer to describe the importance of the input observables: