

(Q)ParticleTransformer application in EW coupling of top quark at CEPC

Huayu Liu

liuhy288@sysu2.edu.cn

School of Science
Sun Yat-sen University

September 23, 2024



- Introduction to the measurement of top quark EW couplings
- Code testing of $t\bar{t}$ jet tagging by ParT/PNet
- Detailed Introduction of Particle Transformer
 - Input data
 - Embedding layer
 - Attention Block layer
 - Class Block layer and output
- Further ParT testing of $t\bar{t}$ event selection by adding variables

1. Introduction to the measurement of top quark EW couplings

1. **Do the traditional method(cut-base)**

Select the $t\bar{t}$ case using some traditional variables.

2. **Do the Machine learning method**

Compared with traditional cut-base ,whether using ML to distinguish $e^+e^- \rightarrow t\bar{t}$ events in CEPC from the background with the same final state is better?

3. **Add quantum part to step2**

4. **Comparison and Summary**

Comprehensive comparison of three methods to evaluate the screening results of machine learning and the model after adding quantum part to $t\bar{t}$ bar

1. **Signal process:** $e^+e^- \rightarrow \gamma/Z \rightarrow t\bar{t}$
2. **Final state:** $l^\pm \nu l^\mp \bar{\nu} b\bar{b}, l^\pm \nu q\bar{q} b\bar{b} \checkmark, \text{all jets}$
 - t quark related information is largely retained in e/u or b-jet.
 - In the chosen final state, the contribution of MET is much more clear than others due to only one neutrino was produced from the process.
3. **Beam state:** 100% of the particles in a beam have a specific chirality, left-handed electrons and right-handed anti-electrons or right-handed electrons and left-handed anti-electrons
4. **Background:** Any process that have $lvqqbb$ final states. The single top production which is difficult to distinguish ($W^* \rightarrow bt$) and others like:
 $\mu\bar{\mu}, q\bar{q}, \gamma/Z, WW, ZZ, ZWW, ZZZ$

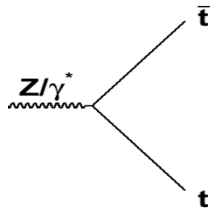


Figure: EW coupling vertex

- If both y/Z and $t\bar{t}$ are on-shell, EW coupling can be described by four Form Factors:

$$\Gamma_u^{Vt\bar{t}}(s) = -ie\{\gamma_u[F_{1V}^X(s) + \gamma_5 F_{1A}^X(s)] + \frac{\sigma_{uv}}{2m_t}(q_t + q_{\bar{t}})[i\frac{F_{2V}^X(s)}{2m_t} + \frac{F_{2A}^X(s)}{e}\gamma_5]\}$$

- Obtained σ_L, σ_R (left/right hand cross section), θ_L, θ_R (polar angle), $(A_{fb}^t)_L, (A_{fb}^t)_R$ (forward backward asymmetry) from the signal process.

They are the are observable which can be measured experimentally.

$$A_{FB}^t = \frac{N(\cos\theta > 0) - N(\cos\theta < 0)}{N(\cos\theta > 0) + N(\cos\theta < 0)}$$

- Four Observables ($\sigma_L, \sigma_R, (A_{fb}^t)_L, (A_{fb}^t)_R$) can be shown by four Form Factors



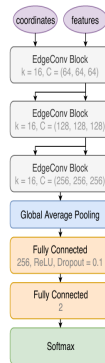
2.Code testing of $t\bar{t}$ jet tagging by ParT/PNet

- **My focus:** Application of machine learning methods in $t\bar{t}$ event selection.
- **Current tasks:**
 1. Test the application of an existing ParT/ParNet code for jet tagging at CEPC.
 2. Further ParT testing of $t\bar{t}$ event selection by adding variables.
- **Simple introduction to ParticleNet/ParticleTransformer:**

1. **ParticleNet** is a model based on GNN architecture that focuses on processing local inter-particle information.

Features: The interaction relationship between particles within a certain range can be captured through the relationship between points and edges.

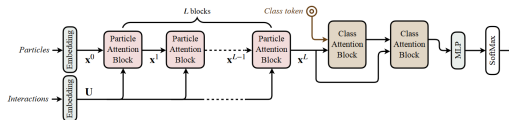
Advantages: Compared with traditional feature engineering and manually defined variables, ParticleNet can automatically learn high-order features of particle injection, improving classification accuracy.



- ParticleTransformer** is a deep learning model that combines self-attention mechanism and interaction between particles.

Features: Ability to flexibly model long-distance dependencies between particles and capture complex relationships between particles and jets.

Advantages: It can better handle the global dependence of particles, and is more robust in identifying jets and background noise.

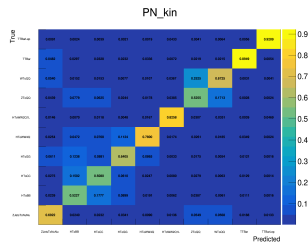
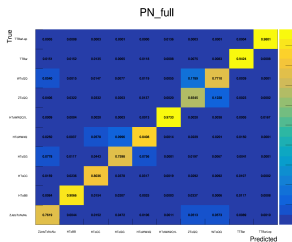
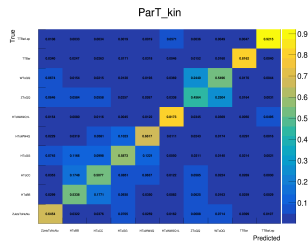
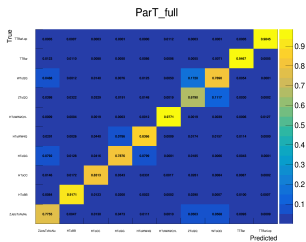


- **Source**(without Quantum part):https://github.com/jet-universe/particle_transformer.
- **input dataset**
10 processes containing jets: $H \rightarrow b\bar{b}, H \rightarrow c\bar{c}, H \rightarrow gg, H \rightarrow 2W \rightarrow 4q, H \rightarrow 2W \rightarrow lvq\bar{q}, t \rightarrow bq\bar{q}, t \rightarrow blv, W \rightarrow q\bar{q}, Z \rightarrow q\bar{q}, Z(j) \rightarrow \bar{\nu}$. (produced by **Shudong**)
Each process has 100M in the training set, 20M in the test set, and 5M in the verification set.
Source: /cefs/higgs/wangshudong/data/JetClass/Pythia
- **features:**
kin: only kinematic variables

$$\Delta\eta, \Delta\phi, \log p_T, \log E, \log \frac{p_T}{p_T(jet)}, \log \frac{E}{E(jet)}, \Delta R$$

full : kinematic variables + particle identification + trajectory displacement

- **epochs:30, Batch Size:1024 , learning rate:0.01**
- **num_heads:8,num_lawyer:8,num_classlayers:2**
- **activation function:gelu**
- **model:ParticleNet/ParticleTransformer**



		ParT_full	ParT_kin	PN_full	PN_kin
CEPC	$\overline{AUC}$	>0.95	>0.95	>0.95	>0.95
	Accuracy	0.85246	0.74387	0.83939	0.6983
JetClass dataset	$\overline{AUC}$	0.9877		0.9849	
	Accuracy	0.861		0.844	

- $\overline{AUC}$ is the average of AUC values between each two processes.

$$\overline{AUC} = \frac{\sum_{i=1}^{10} \sum_{j=1}^{11-i} AUC_{ij}}{\sum_{i=1}^{10} i}$$

- *Accuracy* is the accuracy rate of prediction for the entire test dataset

- There is no significant difference compared to the results in the paper, and there is no significant difference between PN and ParT.
- The results using all features are better than using only kin features.



- 3.Introduction of Particle Transformer
 - Input data

Particle information $\mathbf{x}$

- $\mathbf{x}$ is a tensor of size $[B, F, S]$ describing the particles themselves.
 - B**: The **batch size** which means how many processes are input at one time?
 - F**: means number of features that how many variables describe a particle?
 - S**: The number of particles in the process.

Category	Variable	Definition
Kinematics	$\Delta\eta$	difference in pseudorapidity η between the particle and the jet axis
	$\Delta\phi$	difference in azimuthal angle ϕ between the particle and the jet axis
	$\log p_T$	logarithm of the particle's transverse momentum p_T
	$\log E$	logarithm of the particle's energy
	$\log \frac{p_T}{p_{T(\text{jet})}}$	logarithm of the particle's p_T relative to the jet p_T
	$\log \frac{E}{E(\text{jet})}$	logarithm of the particle's energy relative to the jet energy
	ΔR	angular separation between the particle and the jet axis ($\sqrt{(\Delta\eta)^2 + (\Delta\phi)^2}$)
Particle identification	charge	electric charge of the particle
	Electron	if the particle is an electron ($ pid =11$)
	Muon	if the particle is an muon ($ pid =13$)
	Photon	if the particle is an photon ($pid=22$)
	CH	if the particle is an charged hadron ($ pid =211$ or 321 or 2212)
	NH	if the particle is an neutral hadron ($ pid =130$ or 2112 or 0)
Trajectory displacement	$\tanh d_0$	hyperbolic tangent of the transverse impact parameter value
	$\tanh d_z$	hyperbolic tangent of the longitudinal impact parameter value
	σ_{d_0}	error of the measured transverse impact parameter
	σ_{d_z}	error of the measured longitudinal impact parameter

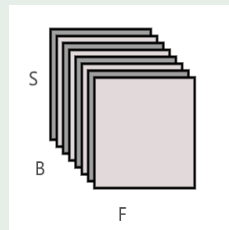


Figure: Example Features

Interaction between particles U

- For one kind of feature, U is a tensor of size [B,C,S,S] which consists of the such following two parts:
- C is the number of the features added to describe the interaction .

The default part which is calculated from p_T of two particles :

$$\Delta_{ij} = \ln \sqrt{\left(\frac{1}{2} \ln\left(1 + \frac{2 * p_{z_i}}{E - p_{z_i}}\right) - \frac{1}{2} \ln\left(1 + \frac{2 * p_{z_j}}{E - p_{z_j}}\right)\right)^2 + \left(\arctan\left(\frac{p_{y_i}}{p_{x_i}}\right)^2 - \arctan\left(\frac{p_{y_j}}{p_{x_j}}\right)^2\right)},$$

$$z_{ij} = \frac{\min(p_{T_i}, p_{T_j})}{p_{T_i} + p_{T_j}},$$

$$k_{Tij} = \min(p_{T_i}, p_{T_j}) * \delta,$$

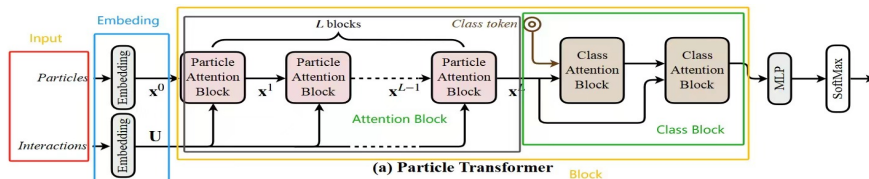
$$m_{ij}^2 = (E_i + E_j)^2 - (p_{T_i} + p_{T_j})^2$$

It can be concatted by artificially added extra matrix uu(size; $[C_u, S, S]$) which is empty by default.

$$C = C_u + 4$$



- 3.Introduction of Particle Transformer
 - Input data
 - Embedding layer



- Through several MLP(Multilayer Perceptron), the feature dimensions of x and U are sequentially raised to the dimensions specified by the $dims$ array.

$$x : [B, F, S] \xrightarrow{MLP} [B, 128, S] \xrightarrow{MLP} [B, 512, S] \xrightarrow{MLP} [B, 128, S] \xrightarrow{Output}$$

$$U : [B, C, S, S] \xrightarrow{MLP} [B, 64, S, S] \xrightarrow{MLP} [B, 64, S, S] \xrightarrow{MLP} [B, 8, S, S] \xrightarrow{Output}$$

- * All numbers are parameters that can be changed.

```
class Embed(nn.Module):
    def __init__(self, input_dim, dims, normalize_input=True, activation='gelu'):
        super().__init__()

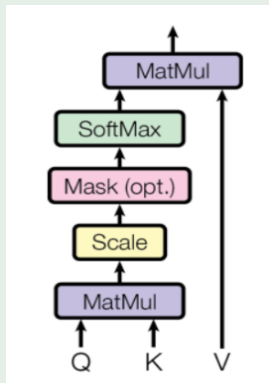
        self.input_bn = nn.BatchNorm1d(input_dim) if normalize_input else None
        module_list = []
        for dim in dims:
            module_list.extend([
                nn.LayerNorm(input_dim),
                nn.Linear(input_dim, dim),
                nn.ReLU() if activation == 'gelu' else nn.ReLU(),
            ])
            input_dim = dim
        self.embed = nn.Sequential(*module_list)

    def forward(self, x):
        if self.input_bn is not None:
            # x: (batch, embed_dim, seq_len)
            x = self.input_bn(x)
            x = x.permute(2, 0, 1).contiguous()
            # x: (seq_len, batch, embed_dim)
            return self.embed(x)
```



- 3.Introduction of Particle Transformer
 - Input data
 - Embedding layer
 - Attention Block layer

Self-Attention (Single Head)



- Calculation of Q(Query), K(Key), V(Value) tensor:

$$\begin{cases} Q = W_q x_n + b_q & \text{size: } [B, N_1, S] \\ K = W_k x_n + b_k & \text{size: } [B, N_1, S] \\ V = W_v x_n + b_v & \text{size: } [B, N_2, S] \end{cases}$$

- Output of *MatMul*:

$$\text{SoftMax} \left(\frac{QK^T}{\sqrt{N_1}} \right) V$$

$$\text{Size: } [B, S, S] * [B, N_2, S] = [B, N_2, S]$$

- Pass Linear layer:

$$x_{n+1} = \text{Linear} \left(\text{SoftMax} \left(\frac{QK^T}{\sqrt{N_1}} \right) V \right)$$

to change dim from N_2 to 128.

Multi-Head

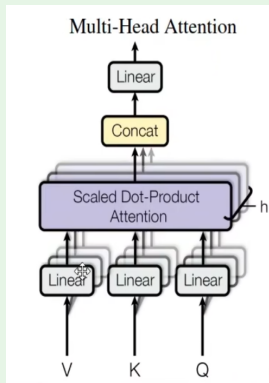
- Divide x into 8 parts on the feature dimension: $x_{n1}, x_{n2}, \dots, x_{n8}$
- Size of x_{ni} : $[B, \frac{128}{8}, N]$
- Calculation of Q (Query), K (Key), V (Value) tensors:

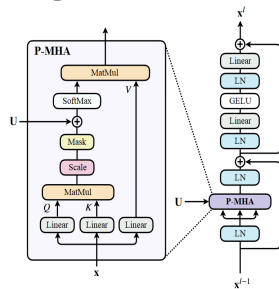
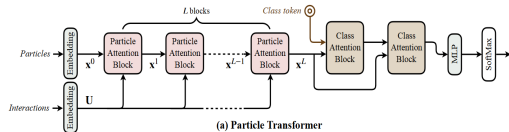
$$\begin{cases} Q_i = W_{qi}x_{ni} + b_{qi} & \text{Size: } [B, N_1, S, 8] \\ K_i = W_{ki}x_{ni} + b_{ki} & \text{Size: } [B, N_1, S, 8] \\ V_i = W_{vi}x_{ni} + b_{vi} & \text{Size: } [B, N_2, S, 8] \end{cases} \quad (1)$$

- Output of *Concat*:

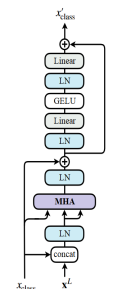
$$\sum_{i=1}^8 \text{SoftMax} \left(\frac{Q_i K_i^T}{\sqrt{N_1}} \right) V_i \oplus$$

$$\text{Size: } [B, 8 \times V, S]$$





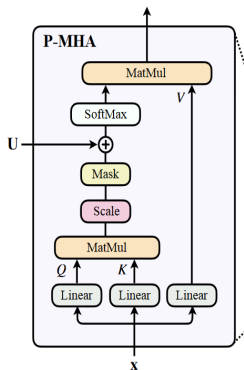
(b) Particle Attention Block



(c) Class Attention Block

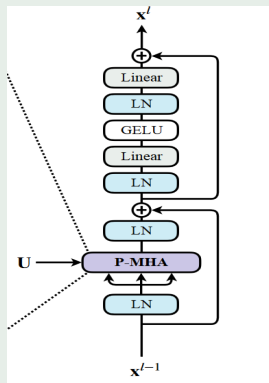
- Here is the complete structure of ParticleTransformer, It can be divided into three modules except the embedding module:
 - Particle Attention Block
 - Class Block
 - Score Calculation
- Here x_{clsaa} is equal to Class token .size[B,128,1]

- The 8 in size of U matrix [B,8,S,S] is required to be equal to head_num. Each head will be allocated a U_i (Size: (B,S,S)).



- * Multi-Head Attention(MHA) Output of $Concat: \sum_{i=1}^8 SoftMax(\frac{Q_i K_i^T}{\sqrt{N_1}}) V_i \oplus$
- * Particle-Multi-Head Attention(P-MHA) Output of $Concat: \sum_{i=1}^8 SoftMax(\frac{Q_i K_i^T}{\sqrt{N_1}} + U_i) V_i \oplus$
- * The size of each part is as same as MHA.

Residual Connection Part



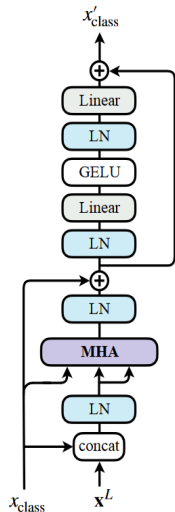
- First Residual Connection:** Set the output of P-MHA as $\tilde{x}_n$, and put $x_n + \tilde{x}_n$ into Residual Connection Part.
- Second Residual Connection:** After $\text{LN}(\text{layer norm})$, Linear layer, Activate function, LN, Linear get $x_n + \tilde{x}_n$.

```
self.pre_fc_norm = nn.LayerNorm(embed_dim)
#LayerNorm
self.fc1 = nn.Linear(embed_dim, self.ffn_dim)
#Linear
self.act = nn.GELU() if activation == 'gelu' else nn.ReLU()
#Activate
self.act_dropout = nn.Dropout(activation_dropout)
#regularization
self.post_fc_norm = nn.LayerNorm(self.ffn_dim) if scale_fc else None
self.fc2 = nn.Linear(self.ffn_dim, embed_dim)
```

- Final output of Block** : $x_{n+1} = x_n + \tilde{x}_n + \tilde{x}_n + \tilde{x}_n$



- 3.Introduction of Particle Transformer
 - Input data
 - Embedding layer
 - Attention Block layer
 - Class Block layer and output



Compare with MHA, the differences of Class Block are;

1. Set $z = (x^{class}, x)$ its size $[B, 128, S+1]$

$$\begin{cases} Q_i = W_{qi}x_{ni}^{class} + b_{qi} & Q \quad \text{size: } [B, N_1, S, 8] \\ K_i = W_{ki}z_{ni} + b_{ki} & K \quad \text{size: } [B, N_1, S, 8] \\ V_i = W_{vi}z_{ni} + b_{vi} & V \quad \text{size: } [B, N_2, S, 8] \end{cases}$$

2. At the first Residual Connection Part replace x_n to x_{class}

3. Output Size: $[B, 128, 1]$

After a MLP (Linear layer) the size is from $[B, 128, 1]$ to $[B, 1, 1]$ which is equivalent to $[B, 1]$.

Then pass the last part and activate it becomes the score of each jet/process.

4. Further ParT testing of $t\bar{t}$ event selection by adding variables

Data: $t\bar{t}$ and SingleTop event level data (sim & reco) from **MUSTAPHA**.
Input Features

Category	Variable	Definition
Particles	$\delta\eta$	difference in pseudorapidity η between the particle and the overall-event axis
	$\delta\phi$	difference in azimuthal angle ϕ between the particle and the overall-event axis
	$\log p_T$	logarithm of the particle' s transverse momentum p_T
	$\log E$	logarithm of the particle' s energy E
	$\log \frac{p_T}{p_T(\text{total})}$	logarithm of the particle' s relative to overall-event $p_T(\text{total})$
	$\log \frac{E}{E(\text{total})}$	logarithm of the particle' s energy relative to the total energy $E(\text{total})$
	δR	angular separation between the particle and the overall-event axis $\sqrt{(\delta\eta)^2 + (\delta\phi)^2}$
Event	JetNum	Jet number of event
	n taus	τ number
	nElec	electron number
	nMuon	μ number
	nGamma	photon number
	E max	the max energy of one object
	MET	Missing Transverse Energy

- Particle features are added by default, while the event features are selected based on significant differences in the distributions of the two types of events as new added features.
- Since the input can only be processed along the particle count dimension, the event features are expanded to match the particle count dimension by replication and concatenated with the particle features for input.

- Missing Transverse Energy

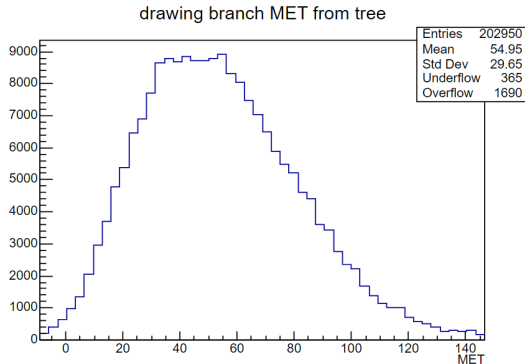


Figure: SingleTopMET

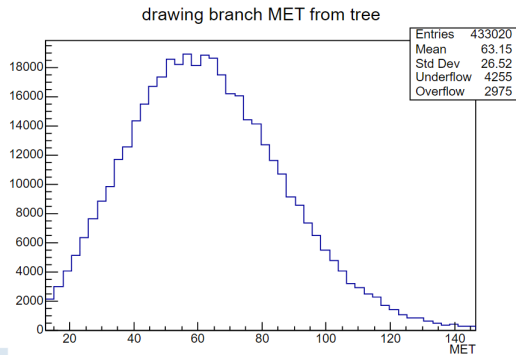


Figure: $t\bar{t}$ MET

- Other features will be shown at backup.

- signal accept score(Input $t\bar{t}$ test data):

drawing branch score_label_sig from Events

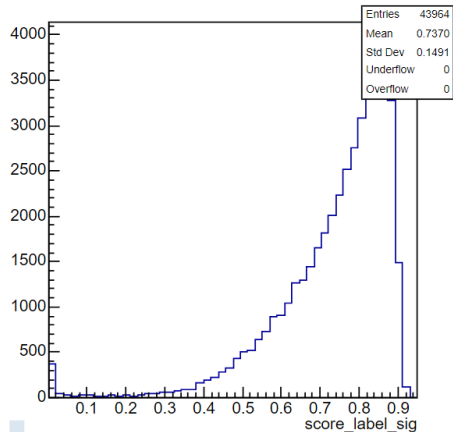


Figure: Just Particle features

drawing branch score_label_sig from Events

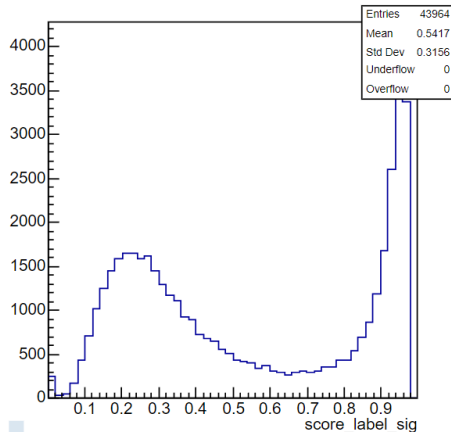


Figure: Particle+Event features

- background reject score(Input SingleTop test data):

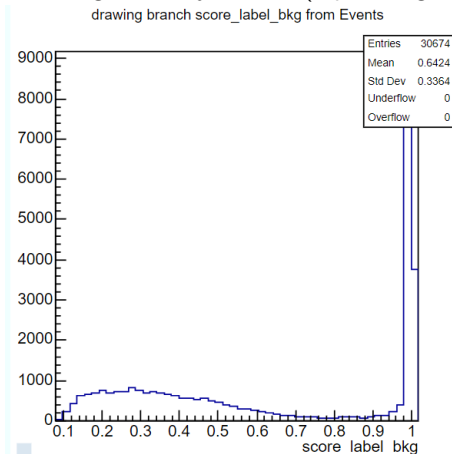


Figure: Just Particle features

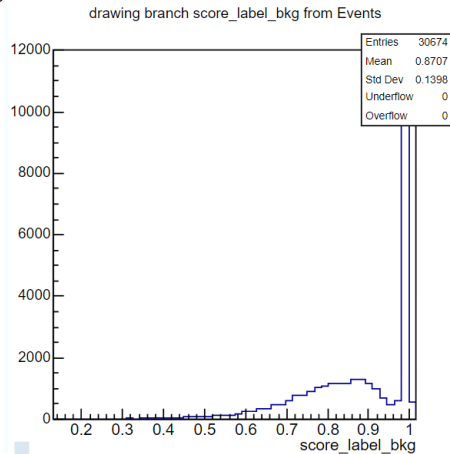


Figure: Particle+Event features

- Accuracy

	Sig	Bkg
Sig	0.9352	0.0647
Bkg	0.4439	0.5570

Just Particle features

	Sig	Bkg
Sig	0.4681	0.5318
Bkg	0.0200	0.9799

Particle+Event features

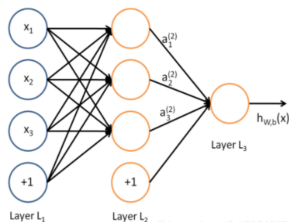
- Suppose there are only these two events, Although the newly added variables do have an effect, they effectively serve as a stricter cut. Generally speaking, the cross-section for $T\bar{T}$ is about 6 to 7 times that of SingleTop. Even though the new model's selection results in fewer events being identified as signals, the proportion of SingleTop background has been significantly reduced.

- Just Particle features:** $\text{signal rate} = \frac{0.9352}{0.9352 + 0.15 \times 0.4439} = 0.9335$

- Particle+Event features:** $\text{signal rate} = \frac{0.4681}{0.4681 + 0.15 \times 0.02} = 0.9936$

1. Use ParT/ParNet to do jet tagging on CEPC MC data. It is verified that parNet and parT can be used for subsequent research.
 - The best bjet identification using traditional methods only has an accuracy of 0.7.
 - It is verified that the two models of PN/ParT are equally effective when applied to CEPC data.
2. Apply the model for $t\bar{t}$ preliminary selection.
3. **Future work plan:**
 - Try to couple charges and add them to the U matrix to describe the electromagnetic interaction between particles.
 - Generate more types of backgrounds ($\mu\bar{\mu}, q\bar{q}, \gamma/Z, WW, ZZ, ZWW, ZZZ$) to verify model optimization results.
 - The current data size is approximately in the range of 50k, and we are trying to obtain more data to increase the stability of the model.

- Code source link:
weaver.ParticleTransformer
- Ref paper:
 1. arXiv:2202.03772v3 [hep-ph] 29 Jan 2024
 2. arXiv:1706.03762v7 [cs.CL] 2 Aug 2023
 3. arXiv:1307.8102v1 [hep-ex] 30 Jul 2013
- Activate function: $GELU(x) \approx 0.5x[1 + \tanh(\sqrt{\frac{2}{\pi}}(x + 0.047715x^3))]$
- MLP/linear:



• NGamma

drawing branch nGamma from tree

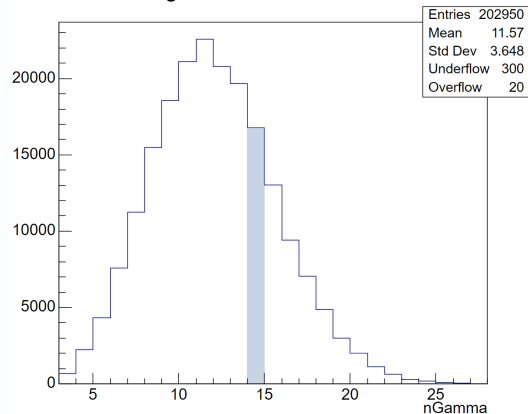
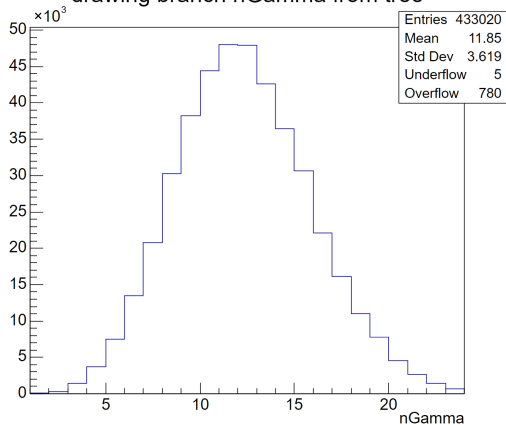


Figure: SingleTopNGamma

drawing branch nGamma from tree

Figure: $t\bar{t}$ NGamma

• Emax

drawing branch Emax from tree

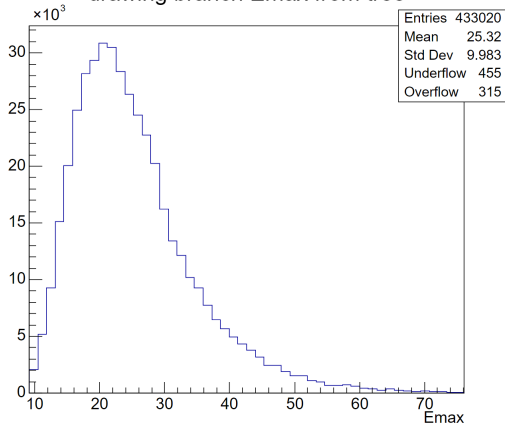
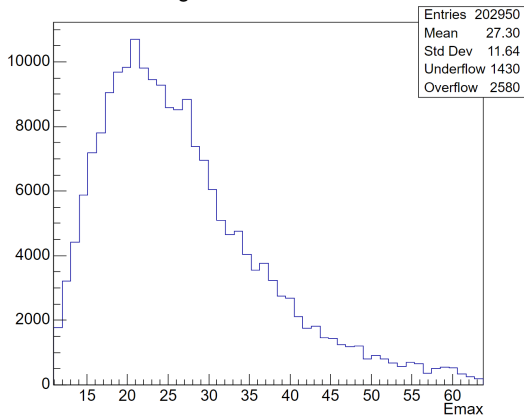


Figure: SingleTopEmax

drawing branch Emax from tree

Figure: $t\bar{t}$ Emax