Note

Feng-Yang Hsieh

1 Higgs Production

We aim to apply deep learning methods to distinguish vector boson fusion (VBF) from gluon-gluon fusion (GGF) and Higgs production at the Large Hadron Collider (LHC).

We aim to apply the CWoLa method, enabling us to utilize real data without prior knowledge of the true label.

2 Sample Preparation

2.1 Monte Carlo samples

We consider Standard Model (SM) di-photon Higgs events produced via GGF and VBF channels at a center-of-mass energy of $\sqrt{s} = 14$ TeV. The Higgs boson events are generated using `MadGraph 3.3.1` [1] for both GGF and VBF production. The Higgs decays into the di-photon final state, and the parton showering and hadronization are simulated using `Pythia 8.306` [2]. The detector simulation is conducted by `Delphes 3.4.2` [3]. Jet reconstruction is performed using `FastJet 3.3.2` [4] with the anti-k_t algorithm [5] and a jet radius of $R = 0.4$. These jets are required to have transverse momentum $p_{\mathrm{T}} > 25$ GeV.

The following `MadGraph` scripts generate Monte Carlo samples for each production channel.

GGF Higgs Sample Generation

```
generate p p > h QCD<=99 [QCD]
output GGF_Higgs
launch GGF_Higgs

shower=Pythia8
detector=Delphes
```

```
analysis=OFF
madspin=OFF
done

set run_card nevents 100000
set run_card ebeam1 7000.0
set run_card ebeam2 7000.0

set run_card use_syst False

set pythia8_card 25:onMode = off
set pythia8_card 25:onIfMatch = 22 22
done
```

VBF Higgs Sample Generation

```
define v = w+ w- z
generate p p > h j j $$v
output VBF_Higgs
launch VBF_Higgs

shower=Pythia8
detector=Delphes
analysis=OFF
madspin=OFF
done

set run_card nevents 100000
set run_card ebeam1 7000.0
set run_card ebeam2 7000.0

set run_card use_syst False

set pythia8_card 25:onMode = off
set pythia8_card 25:onIfMatch = 22 22
done
```

2.2 Event selection

The selection cuts after the `Delphes` simulation:

- n_γ cut: The number of photons should be at least 2.
- n_j cut: The number of jets should be at least 2.
- $m_{\gamma\gamma}$ cut: The invariant mass of two leading photons $m_{\gamma\gamma}$ are required 120 GeV $\leq m_{\gamma\gamma} \leq$ 130 GeV.

Table 1 summarizes the cutflow number at different selection cuts.

Table 1: Number of passing events and passing rates for GGF and VBF Higgs production at different selection cuts.

Cut	GGF	pass rate	VBF	pass rate
Total	100000	1	100000	1
n_γ cut	48286	0.48	53087	0.53
n_j cut	9302	0.09	42860	0.43
$m_{\gamma\gamma}$ cut	8864	0.09	40694	0.41

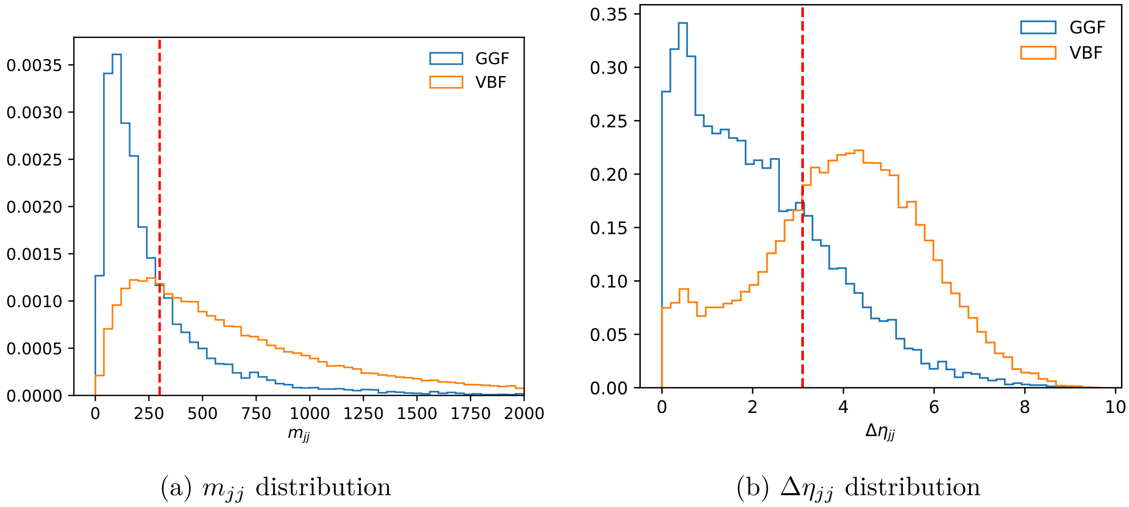
Figure 1 shows the distributions of m_{jj} (the invariant mass of the two leading jets) and $\Delta\eta_{jj}$ (the pseudorapidity difference between the two leading jets). The scatter plot of m_{jj} versus $\Delta\eta_{jj}$ is presented in Figure 2.

2.3 Event image

The inputs for the neural networks are event images [6, 7, 8]. These images are constructed from events that pass the kinematic selection criteria described in section 2.2. Each event image has three channels corresponding to calorimeter towers, tracks, and photons. The following preprocessing steps are applied to all event constituents:

1. Translation: Compute the p_T-weighted center in the ϕ coordinates, then shift this point to the origin.
2. Flipping: Flip the highest p_T quadrant to the first quadrant.
3. Pixelation: Pixelate in a $\eta \in [-5, 5]$, $\phi \in [-\pi, \pi]$ box, with 40×40 pixels

Figure 3 shows the event images for GGF and VBF production modes.



(a) m_{jj} distribution

(b) $\Delta\eta_{jj}$ distribution

Figure 1: Distributions of the invariant mass m_{jj} and pseudorapidity difference $\Delta\eta_{jj}$ of the two leading jets. Red dashed lines are selection cuts used to construct mixed datasets.

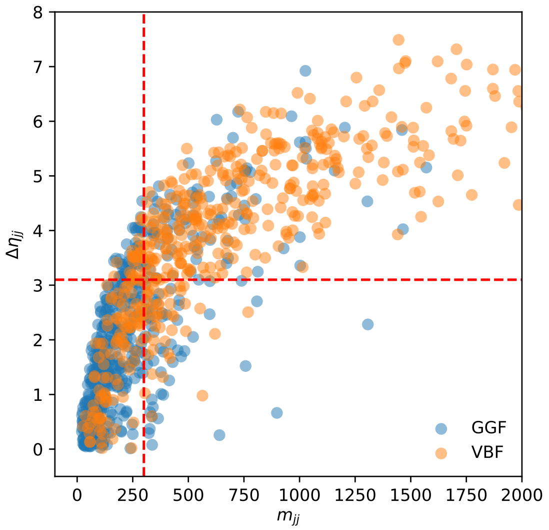
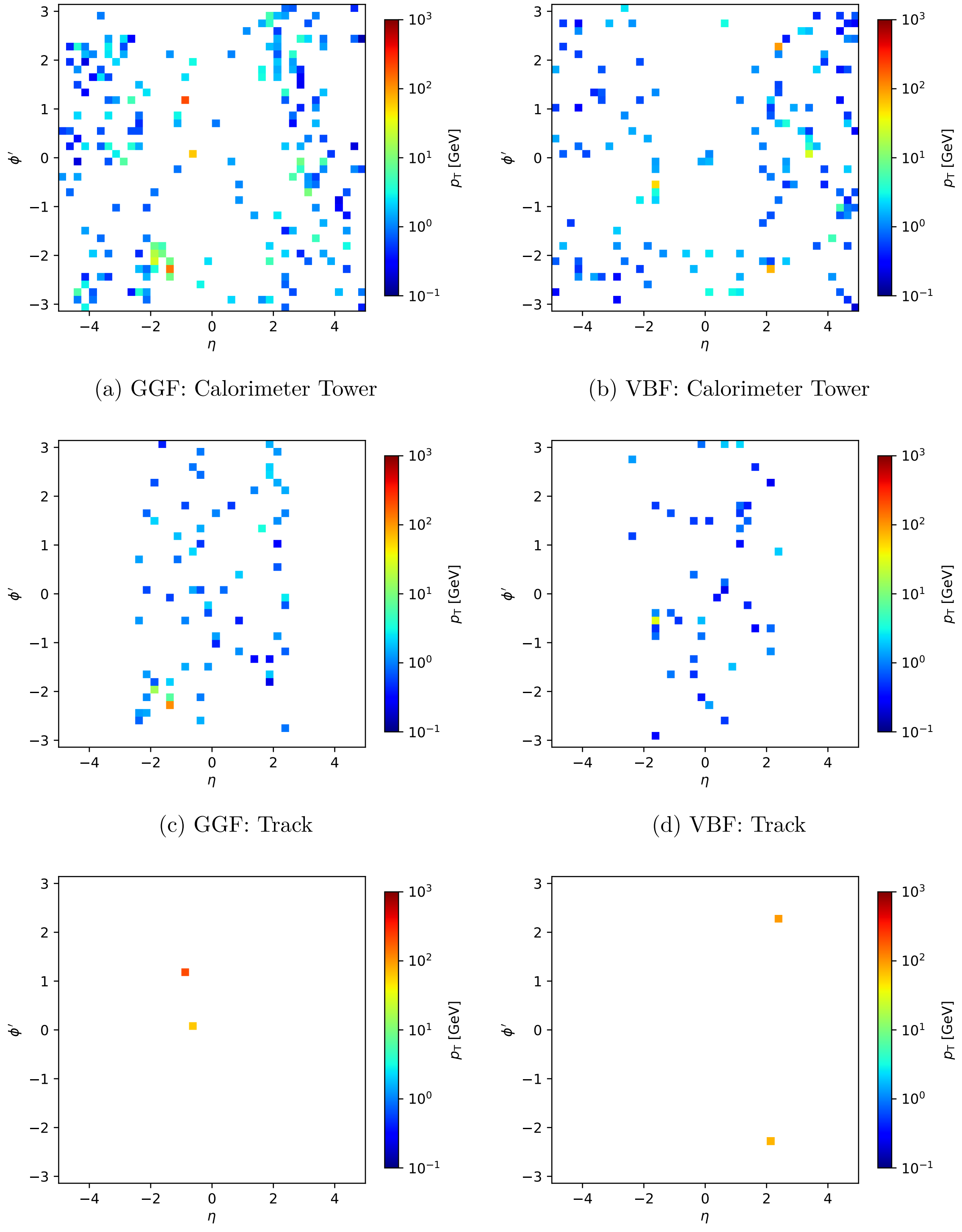


Figure 2: Scatter plot of m_{jj} versus $\Delta\eta_{jj}$. Red dashed lines are selection cuts used to construct mixed datasets.



(a) GGF: Calorimeter Tower

(b) VBF: Calorimeter Tower

(c) GGF: Track

(d) VBF: Track

(e) GGF: Photon

(f) VBF: Photon

Figure 3: Event images for GGF and VBF production, separately shown for calorimeter towers, tracks, and photons.

2.4 Mixed datasets

Based on figure 1, we set selection cuts of $m_{jj} > 300$ GeV and $\Delta\eta_{jj} > 3.1$. We consider three cases: applying each cut individually and simultaneously. These cuts define the signal region (SR), which is VBF-like, and the background region (BR), which is GGF-like. Table 2 summarizes the cutflow results for different selection criteria.

Table 2: Number of passing events and passing rates for GGF and VBF Higgs production under different selection cuts.

Cut	GGF	pass rate	VBF	pass rate
Total	100000	1.00	100000	1.00
n_γ cut	9302	0.09	42860	0.43
n_j cut	9302	0.09	42860	0.43
$m_{\gamma\gamma}$ cut	8864	0.09	40694	0.41
m_{jj} cut: SR	2695	0.03	29496	0.29
m_{jj} cut: BR	6169	0.06	11198	0.11
$\Delta\eta_{jj}$ cut: SR	2317	0.02	28160	0.28
$\Delta\eta_{jj}$ cut: BR	6547	0.07	12534	0.13
$m_{jj}, \Delta\eta_{jj}$ cuts: SR	1832	0.02	26446	0.26
$m_{jj}, \Delta\eta_{jj}$ cuts: BR	5684	0.06	9484	0.09

The total cross-section for VBF production is $\sigma_{\text{VBF}} = 4.278$ pb^{-1} at NNLO and for GGF production is $\sigma_{\text{GGF}} = 54.67$ pb^{-1} at N3LO, as referenced in this link. The branching ratio for the di-photon decay channel is $\Gamma(h \to \gamma\gamma) = 2.270 \times 10^{-3}$, as given in this link.

Assuming the luminosity of $\mathcal{L} = 300$ fb^{-1}, we can estimate the number of events belonging to the SR and BR. These results are summarized in table 3.

3 Training CNN

The total sample sizes are mentioned in section 2.4. We allocate 80% of the data for training and 20% for validation. The testing set consists of the SR's 10,000 VBF and 10,000 GGF events.

The convolutional neural network (CNN) model structure is summarized in figure 4. The internal node uses the rectified linear unit (ReLU) as the activation function. The loss function is the binary cross-entropy. The `Adam` optimizer minimizes the loss value. The learning rate is 10^{-4}, and the batch size is 512. We employ the early stopping technique to

Table 3: The number of events of mixed datasets under different selection cuts.

(a) $m_{jj} > 300$ GeV

	GGF	VBF
BR	2297	326
SR	1003	859

(b) $\Delta\eta_{jj} > 3.1$

	GGF	VBF
BR	2437	365
SR	863	820

(c) $m_{jj} > 300$ GeV,
$\Delta\eta_{jj} > 3.1$

	GGF	VBF
BR	2116	276
SR	682	770

prevent over-training issues with patience of 10.

The training results are summarized in table 4. The performance of the $\Delta\eta_{jj}$ cuts is better than the m_{jj} cut. Moreover, when both cuts are applied together, the performance is slightly worse than when applying either cut individually.

Table 4: The CNN training results. The ACC and AUC are evaluated based on 10 training. The selection cuts of $m_{jj} > 300$ GeV and $\Delta\eta_{jj} > 3.1$ are applied.

	M_1/M_2		S/B	
Cut	ACC	AUC	ACC	AUC
m_{jj}	0.712 ± 0.023	0.741 ± 0.041	0.576 ± 0.010	0.596 ± 0.014
$\Delta\eta_{jj}$	0.828 ± 0.043	0.889 ± 0.050	0.604 ± 0.014	0.630 ± 0.015
$m_{jj}, \Delta\eta_{jj}$	0.753 ± 0.022	0.792 ± 0.035	0.573 ± 0.007	0.596 ± 0.008

3.1 More events

This section assumes the luminosity of $\mathcal{L} = 3000$ fb^{-1}. The number of events belonging to the SR and BR are summarized in table 5.

The training results are summarized in table 6. All datasets' performance is better than the results in table 4. The $\Delta\eta_{jj}$ cut performs better than the m_{jj} cut. Moreover, when both cuts are applied together, the performance is slightly worse than the $\Delta\eta_{jj}$ cut but better than m_{jj}. These results are similar to the previous one.

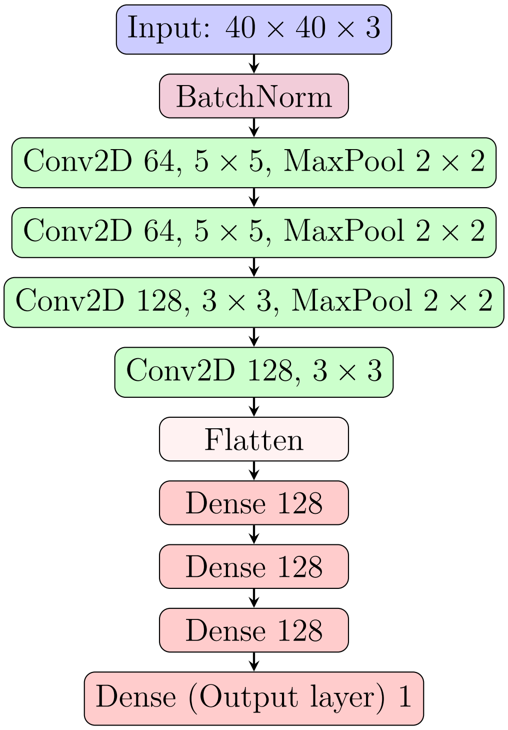


Figure 4: The architecture of the CNN model with key hyperparameters.

Table 5: The number of events of mixed datasets under different selection cuts.

(a) $m_{jj} > 300$ GeV

	GGF	VBF
BR	22967	3262
SR	10034	8593

(b) $\Delta\eta_{jj} > 3.1$

	GGF	VBF
BR	24375	3652
SR	8626	8204

(c) $m_{jj} > 300$ GeV, $\Delta\eta_{jj} > 3.1$

	GGF	VBF
BR	21162	2763
SR	6821	7705

Table 6: The CNN training results. The ACC and AUC are evaluated based on 10 training. The selection cuts of $m_{jj} > 300$ GeV and $\Delta\eta_{jj} > 3.1$ are applied.

	M_1/M_2		S/B	
Cut	ACC	AUC	ACC	AUC
m_{jj}	0.907 ± 0.002	0.969 ± 0.002	0.598 ± 0.008	0.625 ± 0.009
$\Delta\eta_{jj}$	0.931 ± 0.004	0.979 ± 0.002	0.615 ± 0.005	0.648 ± 0.006
$m_{jj}, \Delta\eta_{jj}$	0.929 ± 0.003	0.978 ± 0.002	0.608 ± 0.004	0.638 ± 0.005

4 p_{T} normalization

To remove the potential dependence of the input samples on m_{jj}, we standardize the event images to remove the difference in input data distributions between the SR and BR. We calculate the mean and standard deviation of the event image transverse momentum and use these values to standardize each event image. We standardize each channel separately.

The number of events in the SR and BR are the same as previously in table 5.

The training results are summarized in table 7. The m_{jj} cut performs better than the previous one (table 6).

Table 7: The CNN training results with p_{T} normalization technique. The ACC and AUC are evaluated based on 10 training. The selection cuts of $m_{jj} > 300$ GeV and $\Delta\eta_{jj} > 3.1$ are applied.

	M_1/M_2		S/B	
Cut	ACC	AUC	ACC	AUC
m_{jj}	0.874 ± 0.004	0.946 ± 0.003	0.624 ± 0.005	0.663 ± 0.006
$\Delta\eta_{jj}$	0.928 ± 0.005	0.979 ± 0.002	0.597 ± 0.005	0.630 ± 0.006
$m_{jj}, \Delta\eta_{jj}$	0.917 ± 0.003	0.973 ± 0.002	0.603 ± 0.004	0.636 ± 0.006

5 Different cut setting

We set selection cuts of $m_{jj} > 225$ GeV and $\Delta\eta_{jj} > 2.3$ to ensure the SR and BR datasets have similar sizes. Table 8 summarizes the cutflow results for different selection criteria.

Assuming the luminosity of $\mathcal{L} = 3000$ fb^{-1}, we can estimate the number of events belonging to the SR and BR. These results are summarized in table 9

Table 8: Number of passing events and passing rates for GGF and VBF Higgs production under different selection cuts.

Cut	GGF	pass rate	VBF	pass rate
Total	100000	1.00	100000	1.00
n_γ cut	9302	0.09	42860	0.43
n_j cut	9302	0.09	42860	0.43
$m_{\gamma\gamma}$ cut	8864	0.09	40694	0.41
m_{jj} cut: SR	3638	0.04	32993	0.33
m_{jj} cut: BR	5226	0.05	7701	0.08
$\Delta\eta_{jj}$ cut: SR	3611	0.04	32914	0.33
$\Delta\eta_{jj}$ cut: BR	5253	0.05	7780	0.08
$m_{jj}, \Delta\eta_{jj}$ cuts: SR	2842	0.03	31113	0.31
$m_{jj}, \Delta\eta_{jj}$ cuts: BR	4457	0.04	5900	0.06

Table 9: The number of events of mixed datasets under different selection cuts.

(a) $m_{jj} > 225$ GeV

	GGF	VBF
BR	19457	2244
SR	13544	9612

(b) $\Delta\eta_{jj} > 2.3$

	GGF	VBF
BR	19557	2267
SR	13444	9589

(c) $m_{jj} > 225$ GeV, $\Delta\eta_{jj} > 2.3$

	GGF	VBF
BR	16594	1719
SR	10581	9064

The training results are summarized in table 10. The results are better than the table 7 by 1%. Similarly, the m_{jj} cut performs best.

Table 10: The CNN training results with p_{T} normalization technique. The ACC and AUC are evaluated based on 10 training. The selection cuts of $m_{jj} > 225$ GeV and $\Delta\eta_{jj} > 2.3$ are applied.

	M_1/M_2		S/B	
Cut	ACC	AUC	ACC	AUC
m_{jj}	0.864 ± 0.004	0.940 ± 0.004	0.632 ± 0.006	0.673 ± 0.007
$\Delta\eta_{jj}$	0.913 ± 0.006	0.972 ± 0.003	0.605 ± 0.007	0.640 ± 0.009
$m_{jj}, \Delta\eta_{jj}$	0.896 ± 0.007	0.961 ± 0.004	0.616 ± 0.005	0.653 ± 0.006

6 Supervised training

This section tests the supervised training on CNN. The training, validation, and testing sample sizes are summarized in table 11. The events passing all selection requirements (section 2.2) are considered.

Table 11: Sizes of various samples used for supervised training.

	Training	Validation	Testing
GGF	100k	25k	25k
VBF	100k	25k	25k

The training results are summarized in table 12. These results demonstrate the upper limit of CNN training.

Table 12: The CNN training results with p_{T} normalization technique. The ACC and AUC are evaluated based on 10 training.

ACC	AUC
0.784 ± 0.001	0.861 ± 0.001

6.1 Testing sample in SR and BR

The testing events used to evaluate the table 12 are all events passing the selection and not restricted to the particular SR. Thus, to make a fair comparison with previous results, we must evaluate the training performance on the events in SR and BR.

The new testing dataset consists of the 10,000 VBF and 10,000 GGF events from SR and BR. The numbers of SR and BR events are computed from table 8.

The training results of table 10 are re-evaluated on the new testing set and shown in table 13. The results are better than the table 10. It seems that the events in the BR can be distinguished better than those in the SR.

Table 13: The CNN training results with $p_{\rm T}$ normalization technique. The ACC and AUC are evaluated based on 10 training. The selection cuts of $m_{jj} > 225$ GeV and $\Delta\eta_{jj} > 2.3$ are applied.

	M_1/M_2		S/B	
Cut	ACC	AUC	ACC	AUC
m_{jj}	0.863 ± 0.004	0.940 ± 0.002	0.716 ± 0.003	0.780 ± 0.004
$\Delta\eta_{jj}$	0.914 ± 0.004	0.972 ± 0.003	0.702 ± 0.003	0.754 ± 0.003
$m_{jj}, \Delta\eta_{jj}$	0.896 ± 0.006	0.962 ± 0.004	0.723 ± 0.003	0.780 ± 0.002

7 Use jet tagging results to construct mixed datasets

This section uses the jet tagging results to construct the mixed datasets.

Assuming the luminosity of $\mathcal{L} = 3000$ fb^{-1}, we can estimate the number of events belonging to the SR and BR. The SR and BR are defined based on the number of gluon jets n_g and quark jets n_q. The selection results are summarized in table 14.

Table 14: The number of events of mixed datasets under different selection cuts. Here, $agbq$ means that $n_g = a, n_q = b$.

(a) SR: $2q0g$;
BR: $1q1g, 0q2g$

	GGF	VBF
SR	16828	10229
BR	16865	1596

(b) SR: $2q0g, 1q1g$;
BR: $0q2g$

	GGF	VBF
SR	30752	11779
BR	2941	47

(c) SR: $2q0g$; BR: $0q2g$

	GGF	VBF
SR	16828	10229
BR	2941	47

For now, we use the true information from `Delphes` and do not consider the mis-tagging case.

The training results are summarized in table 15. All different jet-tagging conditions produced similar performance. However, the results are worse than those of kinematic cuts (table 13).

Table 15: The CNN training results with p_{T} normalization technique. The ACC and AUC are evaluated based on 10 training.

	M_1/M_2		S/B	
Datasets	ACC	AUC	ACC	AUC
SR: $2q0g$; BR: $1q1g, 0q2g$	0.623 ± 0.005	0.642 ± 0.005	0.653 ± 0.008	0.706 ± 0.009
SR: $2q0g, 1q1g$; BR: $0q2g$	0.934 ± 0.000	0.689 ± 0.012	0.662 ± 0.006	0.719 ± 0.008
SR: $2q0g$; BR: $0q2g$	0.900 ± 0.000	0.740 ± 0.010	0.655 ± 0.008	0.710 ± 0.009

The training results without p_{T} nomalization are summarized in table 16. All different jet-tagging conditions produced similar performance. However, the results are worse than the ones with p_{T} normalization (table 15) by 2%.

Table 16: The CNN training results without p_{T} normalization technique. The ACC and AUC are evaluated based on 10 training.

	M_1/M_2		S/B	
Datasets	ACC	AUC	ACC	AUC
SR: $2q0g$; BR: $1q1g, 0q2g$	0.614 ± 0.007	0.632 ± 0.011	0.646 ± 0.008	0.690 ± 0.011
SR: $2q0g, 1q1g$; BR: $0q2g$	0.934 ± 0.000	0.695 ± 0.015	0.643 ± 0.009	0.689 ± 0.011
SR: $2q0g$; BR: $0q2g$	0.900 ± 0.000	0.743 ± 0.011	0.632 ± 0.007	0.677 ± 0.008

7.1 Loss weighted

Since the sample sizes are unbalanced, we add the class weights. The weights are proportional to the reciprocal of the number of events.

The training results with class weights are summarized in table 17. All different jet-tagging conditions produced similar performance.

8 Total scaling of transverse momentum

The p_{T} normalization removes the magnitude information of the input datasets. Thus, we would expect the training performance of the p_{T} normalization datasets would be worse than the one without it. However, table 15 and 16 shows the opposite results.

Table 17: The CNN training results without p_{T} normalization technique. The ACC and AUC are evaluated based on 10 training.

	M_1/M_2		S/B	
Datasets	ACC	AUC	ACC	AUC
SR: $2q0g$; BR: $1q1g, 0q2g$	0.621 ± 0.006	0.635 ± 0.007	0.645 ± 0.009	0.688 ± 0.013
SR: $2q0g, 1q1g$; BR: $0q2g$	0.934 ± 0.000	0.679 ± 0.016	0.624 ± 0.005	0.662 ± 0.008
SR: $2q0g$; BR: $0q2g$	0.900 ± 0.000	0.730 ± 0.013	0.621 ± 0.005	0.658 ± 0.008

To explore the reason why the p_{T} normalization could improve the training performance, we try the total p_{T} scaling, which computes the mean and standard deviation of all input samples. Then, use these values to standardize the input datasets.

8.1 Results

The training results with p_{T} scaling are summarized in table 18. All different jet-tagging conditions produced similar performance. However, the results are worse than the ones with p_{T} normalization (table 15).

Table 18: The CNN training results with p_{T} scaling technique. The ACC and AUC are evaluated based on 10 training. The selection cuts on the number of gluon jets are applied.

	M_1/M_2		S/B	
Datasets	ACC	AUC	ACC	AUC
SR: $2q0g$; BR: $1q1g, 0q2g$	0.622 ± 0.004	0.637 ± 0.008	0.638 ± 0.009	0.678 ± 0.011
SR: $2q0g, 1q1g$; BR: $0q2g$	0.934 ± 0.000	0.673 ± 0.032	0.619 ± 0.019	0.652 ± 0.029
SR: $2q0g$; BR: $0q2g$	0.900 ± 0.000	0.733 ± 0.011	0.621 ± 0.006	0.657 ± 0.009

The training results with p_{T} normalization are summarized in table 19.

The training results without p_{T} normalization are summarized in table 20.

9 Data augmentation

To improve the training performance, we will consider various data augmentation methods.

Table 19: The CNN training results with p_{T} normalization technique. The ACC and AUC are evaluated based on 10 training. The selection cuts on the number of gluon jets are applied.

	M_1/M_2		S/B	
Datasets	ACC	AUC	ACC	AUC
SR: $2q0g$; BR: $1q1g, 0q2g$	0.615 ± 0.005	0.632 ± 0.007	0.650 ± 0.011	0.703 ± 0.015
SR: $2q0g, 1q1g$; BR: $0q2g$	0.934 ± 0.000	0.662 ± 0.014	0.630 ± 0.008	0.675 ± 0.011
SR: $2q0g$; BR: $0q2g$	0.900 ± 0.000	0.716 ± 0.012	0.640 ± 0.007	0.690 ± 0.009

Table 20: The CNN training results without p_{T} normalization technique. The ACC and AUC are evaluated based on 10 training. The selection cuts on the number of gluon jets are applied.

	M_1/M_2		S/B	
Datasets	ACC	AUC	ACC	AUC
SR: $2q0g$; BR: $1q1g, 0q2g$	0.620 ± 0.004	0.636 ± 0.005	0.643 ± 0.006	0.686 ± 0.007
SR: $2q0g, 1q1g$; BR: $0q2g$	0.934 ± 0.000	0.680 ± 0.014	0.624 ± 0.010	0.660 ± 0.016
SR: $2q0g$; BR: $0q2g$	0.900 ± 0.000	0.727 ± 0.010	0.628 ± 0.008	0.666 ± 0.011

9.1 p_{T} smearing

The p_{T} smearing method simulates detector resolution effects on the transverse momentum of event constituents. This method resamples the transverse momentum p_{T} of event constituents according to the normal distribution:

$$p'_{\mathrm{T}} \sim \mathcal{N}\left(p_{\mathrm{T}}, f(p_{\mathrm{T}})\right), \quad f(p_{\mathrm{T}}) = \sqrt{0.052 p_{\mathrm{T}}^2 + 1.502 p_{\mathrm{T}}}, \tag{1}$$

where p'_{T} is the augmented transverse momentum, and $f\left(p_{\mathrm{T}}\right)$ is the energy smearing function applied by `Delphes` (the p_{T}'s are normalized in units of GeV). The preprocessing is applied after the p_{T} smearing augmentation.

The training results of the $2q0g$ datasets (Table 14 (a)) are summarized in table 21.

9.2 ϕ shifting

The ϕ shifting method shifts entire events by a random angle $\Delta\phi \in [-\pi, \pi]$ to enlarge the diversity of training datasets.

The training results of the $2q0g$ datasets are summarized in table 22.

Table 21: CNN training results with different augmentation sizes. The ACC and AUC are evaluated based on 10 training.

	M_1/M_2		S/B	
Datasets	ACC	AUC	ACC	AUC
Original	0.615 ± 0.005	0.632 ± 0.007	0.650 ± 0.011	0.703 ± 0.015
+5	0.625 ± 0.006	0.653 ± 0.009	0.661 ± 0.010	0.714 ± 0.012
+10	0.629 ± 0.005	0.658 ± 0.005	0.666 ± 0.008	0.721 ± 0.009
+15	0.629 ± 0.003	0.660 ± 0.003	0.661 ± 0.015	0.710 ± 0.018

Table 22: CNN training results with different augmentation sizes. The ACC and AUC are evaluated based on 10 training.

	M_1/M_2		S/B	
Datasets	ACC	AUC	ACC	AUC
Original	0.615 ± 0.005	0.632 ± 0.007	0.650 ± 0.011	0.703 ± 0.015
+5	0.641 ± 0.003	0.680 ± 0.004	0.683 ± 0.010	0.736 ± 0.013
+10	0.642 ± 0.006	0.684 ± 0.008	0.686 ± 0.008	0.739 ± 0.011
+15	0.643 ± 0.005	0.685 ± 0.006	0.687 ± 0.009	0.742 ± 0.010

9.3 $\eta - \phi$ smearing

We apply the $\eta - \phi$ smearing on the training samples. Specifically, the (η, ϕ) coordinates of constituents are resampled according to a normal distribution centered on the original coordinate and with a standard deviation inversely proportional to the p_{T}

$$\eta' \sim \mathcal{N}\left(\eta, \frac{\Lambda}{p_{\mathrm{T}}}\right), \quad \phi' \sim \mathcal{N}\left(\phi, \frac{\Lambda}{p_{\mathrm{T}}}\right) \tag{2}$$

where η', ϕ' are the augmented coordinates, p_{T} is the transverse momentum of the constituent, and the smearing scale is set to be $\Lambda = 100$ MeV.

The training results on the $2q0g$ datasets are summarized in Table 23. The +5 and +10 augmentation cases show performance comparable to the original dataset. However, applying +15 augmentations degrades the performance, suggesting that introducing too many augmented samples may lead the training in the wrong direction.

9.4 Without pre-processing

The ϕ shifting seems to cancel the ϕ translation in the pre-processing. Thus, we expect the model trained on the ϕ shifting dataset could perform similarly to the no pre-processing

Table 23: CNN training results with different augmentation sizes. The ACC and AUC are evaluated based on 10 training.

	M_1/M_2		S/B	
Datasets	ACC	AUC	ACC	AUC
Original	0.615 ± 0.005	0.632 ± 0.007	0.650 ± 0.011	0.703 ± 0.015
+5	0.618 ± 0.004	0.640 ± 0.006	0.658 ± 0.009	0.711 ± 0.013
+10	0.617 ± 0.004	0.641 ± 0.006	0.654 ± 0.010	0.705 ± 0.012
+15	0.612 ± 0.006	0.628 ± 0.008	0.635 ± 0.009	0.679 ± 0.013

datasets.

The testing results of the $2q0g$ datasets are summarized in table 24. The performance of pre-processing datasets is generally better than that without pre-processing. The reason may be that the original datasets are applied pre-processed. Thus, the samples have higher density for the ϕ center at 0. The model would prefer to learn these events first.

We can train the model on only the augmented datasets to ensure the effect of the original samples.

Table 24: CNN training results with different augmentation sizes. The ACC and AUC are evaluated based on 10 training.

	w/ pre-processing		w/o pre-processing	
Datasets	ACC	AUC	ACC	AUC
Original	0.637 ± 0.007	0.686 ± 0.008	0.625 ± 0.006	0.669 ± 0.008
+5	0.682 ± 0.011	0.735 ± 0.013	0.669 ± 0.011	0.720 ± 0.015
+10	0.685 ± 0.008	0.739 ± 0.010	0.673 ± 0.008	0.726 ± 0.010
+15	0.688 ± 0.007	0.743 ± 0.009	0.674 ± 0.008	0.726 ± 0.008

9.5 Only augmentation datasets

In section 9.4, we found that the performance of pre-processing datasets is generally better than without pre-processing. We train the model on only the augmented datasets to ensure the effect of the original samples.

The testing results of the only augmented sample are summarized in table 25. The performance without original samples is similar to that with original samples. It seems that the impact of the original datasets is limited. For 10, 15 augmentation cases, with and without original samples perform almost the same.

Table 25: CNN training results with different augmentation sizes. The ACC and AUC are evaluated based on 10 training. Here, $+x$ contains original and augmented samples; $=x$ contains only augmented samples.

	w/ pre-processing		w/o pre-processing	
Datasets	ACC	AUC	ACC	AUC
+5	0.682 ± 0.011	0.735 ± 0.013	0.669 ± 0.011	0.720 ± 0.015
=5	0.682 ± 0.007	0.736 ± 0.009	0.668 ± 0.007	0.718 ± 0.010
+10	0.685 ± 0.008	0.739 ± 0.010	0.673 ± 0.008	0.726 ± 0.010
=10	0.687 ± 0.010	0.740 ± 0.012	0.675 ± 0.009	0.726 ± 0.011
+15	0.688 ± 0.007	0.743 ± 0.009	0.674 ± 0.008	0.726 ± 0.008
=15	0.687 ± 0.007	0.741 ± 0.010	0.672 ± 0.009	0.725 ± 0.012

10 Removing photon information

To investigate the role of photon information in model training, we conduct two exercises:

- **Case 1:** Remove both the photon channel and photon features in the Tower channel.
- **Case 2:** Remove the photon channel.
- **Case 3:** Remove the photon features in the Tower channel.

We consider the $2q0g$ datasets. The training results are summarized in Tables 26, 27, and 28.

Table 26: CNN training results with both the photon channel and Tower photon features removed (Case 1). The ACC and AUC are evaluated based on 10 training.

	M_1/M_2		S/B	
Datasets	ACC	AUC	ACC	AUC
Original	0.633 ± 0.005	0.664 ± 0.008	0.690 ± 0.005	0.750 ± 0.008
+5	0.644 ± 0.004	0.687 ± 0.005	0.693 ± 0.007	0.746 ± 0.006
+10	0.645 ± 0.004	0.689 ± 0.005	0.697 ± 0.009	0.751 ± 0.011
+15	0.645 ± 0.004	0.689 ± 0.005	0.698 ± 0.009	0.753 ± 0.010

Case 1 leads the performance improvement compared to the full-feature baseline (Table 22). Case 2 shows a little better performance of the all-feature input case. Case 3 demonstrates more improvement than case 2 but is still worse than case 1.

Table 27: CNN training results with only the photon channel removed (Case 2). The ACC and AUC are evaluated based on 10 training.

	M_1/M_2		S/B	
Datasets	ACC	AUC	ACC	AUC
Original	0.621 ± 0.005	0.640 ± 0.008	0.661 ± 0.007	0.715 ± 0.010
+5	0.636 ± 0.004	0.673 ± 0.006	0.673 ± 0.009	0.727 ± 0.011
+10	0.639 ± 0.004	0.677 ± 0.005	0.677 ± 0.007	0.728 ± 0.010
+15	0.640 ± 0.005	0.679 ± 0.007	0.678 ± 0.007	0.731 ± 0.010

Table 28: CNN training results with the Tower photon features removed (Case 3). The ACC and AUC are evaluated based on 10 training.

	M_1/M_2		S/B	
Datasets	ACC	AUC	ACC	AUC
Original	0.629 ± 0.004	0.653 ± 0.006	0.670 ± 0.008	0.726 ± 0.009
+5	0.642 ± 0.005	0.682 ± 0.008	0.690 ± 0.010	0.742 ± 0.011
+10	0.644 ± 0.003	0.686 ± 0.004	0.692 ± 0.006	0.746 ± 0.008
+15	0.645 ± 0.005	0.687 ± 0.007	0.690 ± 0.007	0.745 ± 0.007

11 Various decay channels

In this section, we consider $H \to W^+W^-$, $H \to ZZ$, and $H \to \tau^+\tau^-$.

11.1 Final state

Unlike the $H \to \gamma\gamma$ analysis, the final states of $H \to WW^*$, ZZ^*, and $\tau^+\tau^-$ are more diverse. Each decay mode offers several final-state configurations, depending on whether the intermediate particles decay leptonically or hadronically.

11.1.1 $H \to WW^*$

- **Fully leptonic:** $H \to WW^* \to \ell\nu\ell\nu$
- **Semi-leptonic:** $H \to WW^* \to \ell\nu jj$
- **Fully hadronic:** $H \to WW^* \to jjjj$

11.1.2 $H \to ZZ^*$

- **Fully leptonic:** $H \to ZZ^* \to 4\ell$
- **Semi-leptonic:** $H \to ZZ^* \to 2\ell 2j$
- **Invisible + leptonic:** $H \to ZZ^* \to 2\ell 2\nu$
- **Fully hadronic:** $H \to ZZ^* \to 4j$

11.1.3 $H \to \tau^+\tau^-$

- **Leptonic–Leptonic:** $H \to \tau^+\tau^- \to \ell\nu\bar{\nu}\ \ell\nu\bar{\nu}$
- **Leptonic–Hadronic:** $H \to \tau^+\tau^- \to \ell\nu\bar{\nu}\ \tau_{\text{had}}$
- **Hadronic–Hadronic:** $H \to \tau^+\tau^- \to \tau_{\text{had}}\tau_{\text{had}}$

11.2 Pre-selection cuts

Each decay mode requires tailored pre-selection cuts to suppress background while retaining a reasonable signal efficiency. Some considerations:

- How should pre-selection cuts be defined for each decay channel?
- Should we unify pre-selection criteria to allow CNN to transfer across channels?

Answers:

- Consider the **Fully leptonic:** $H \to ZZ^* \to 4\ell$ channel. Only require the number of leptons and jets.
- The invariant mass cuts can be removed.

12 ϕ shifting with fixed angle

In section 9.2, we implemented ϕ-shifting augmentation using a random rotation angle for each event. In this section, we consider shifting the azimuthal angle ϕ by fixed values and investigate its effect on training performance.

The training results on the $2q0g$ dataset are summarized in table 29. We find that using fixed-angle ϕ shifts yields performance comparable to the random-angle augmentation results shown previously in table 22.

Table 29: CNN training results using fixed-angle ϕ-shifting augmentation. Here, $360/\theta$ denotes that events were augmented using rotation angles of θ, 2θ, ..., up to 360°. The ACC and AUC are evaluated based on 10 training.

	M_1/M_2		S/B	
Datasets	ACC	AUC	ACC	AUC
Original	0.615 ± 0.005	0.632 ± 0.007	0.650 ± 0.011	0.703 ± 0.015
360/90	0.637 ± 0.005	0.675 ± 0.007	0.684 ± 0.010	0.739 ± 0.012
360/60	0.642 ± 0.003	0.682 ± 0.005	0.689 ± 0.007	0.745 ± 0.010
360/45	0.643 ± 0.003	0.685 ± 0.004	0.689 ± 0.008	0.742 ± 0.012
360/30	0.643 ± 0.006	0.684 ± 0.009	0.688 ± 0.007	0.744 ± 0.008

13 $H \to ZZ^* \to 4\ell$ channel

13.1 Sample preparation

We consider SM Higgs decay into ZZ via GGF and VBF channels at a center-of-mass energy of $\sqrt{s} = 14$ TeV. We focus on the fully leptonic mode: $H \to ZZ^* \to 4\ell$. The Higgs boson events are generated using MadGraph 3.3.1 [1] for both GGF and VBF production. The parton showering and hadronization are simulated using Pythia 8.306 [2]. The detector simulation is conducted by Delphes 3.4.2 [3]. Jet reconstruction is performed using FastJet 3.3.2 [4] with the anti-k_t algorithm [5] and a jet radius of $R = 0.4$. These jets are required to have transverse momentum $p_{\mathrm{T}} > 25$ GeV.

The following MadGraph scripts generate Monte Carlo samples for each production channel.

GGF Higgs Sample Generation

```
import model loop_sm
generate p p > h > l+ l- l+ l- QCD=0 QED<=4 [noborn=QCD]
output GGF_Higgs_ZZ_4l
launch GGF_Higgs_ZZ_4l

shower=Pythia8
detector=Delphes
analysis=OFF
madspin=OFF
done
```

```
Cards/delphes_card.dat

set run_card nevents 10000
set run_card ebeam1 7000.0
set run_card ebeam2 7000.0

set run_card use_syst False

done
```

VBF Higgs Sample Generation

```
define v = w+ w- z
generate p p > h j j $$v, (h > z z , z > l+ l- , z > l+ l-) QCD<=99
output VBF_Higgs_ZZ_4l
launch VBF_Higgs_ZZ_4l

shower=Pythia8
detector=Delphes
analysis=OFF
madspin=OFF
done

Cards/delphes_card.dat

set run_card nevents 10000
set run_card ebeam1 7000.0
set run_card ebeam2 7000.0

set run_card use_syst False

done
```

The selection cuts after the `Delphes` simulation:

- n_l cut: The number of leptons should be at least 4.

- n_j cut: The number of jets should be at least 2.

Table 30 summarizes the cutflow number at different selection cuts.

Table 30: Number of passing events and passing rates for GGF and VBF Higgs production at different selection cuts.

Cut	GGF	pass rate	VBF	pass rate
Total	10000	1	10000	1
n_l cut	2731	0.27	1902	0.19
n_j cut	687	0.07	1650	0.17

The branching ratio for the 4-lepton channel is $\Gamma\left(h \to 4\ell, \ell = e, \mu\right) = 1.240 \times 10^{-4}$, as given in this link. Assuming the luminosity of $\mathcal{L} = 3000\ \text{fb}^{-1}$, we can estimate the number of events belonging to the SR and BR. The SR and BR are defined based on the number of gluon jets n_g and quark jets n_q. The selection results are summarized in table 31.

Table 31: The number of events of mixed datasets under different selection cuts. Here, $agbq$ means that $n_g = a, n_q = b$.

(a) SR: $2q0g$;
BR: $1q1g, 0q2g$

	GGF	VBF
SR	722	228
BR	704	34

(b) SR: $2q0g, 1q1g$;
BR: $0q2g$

	GGF	VBF
SR	1287	261
BR	138	1

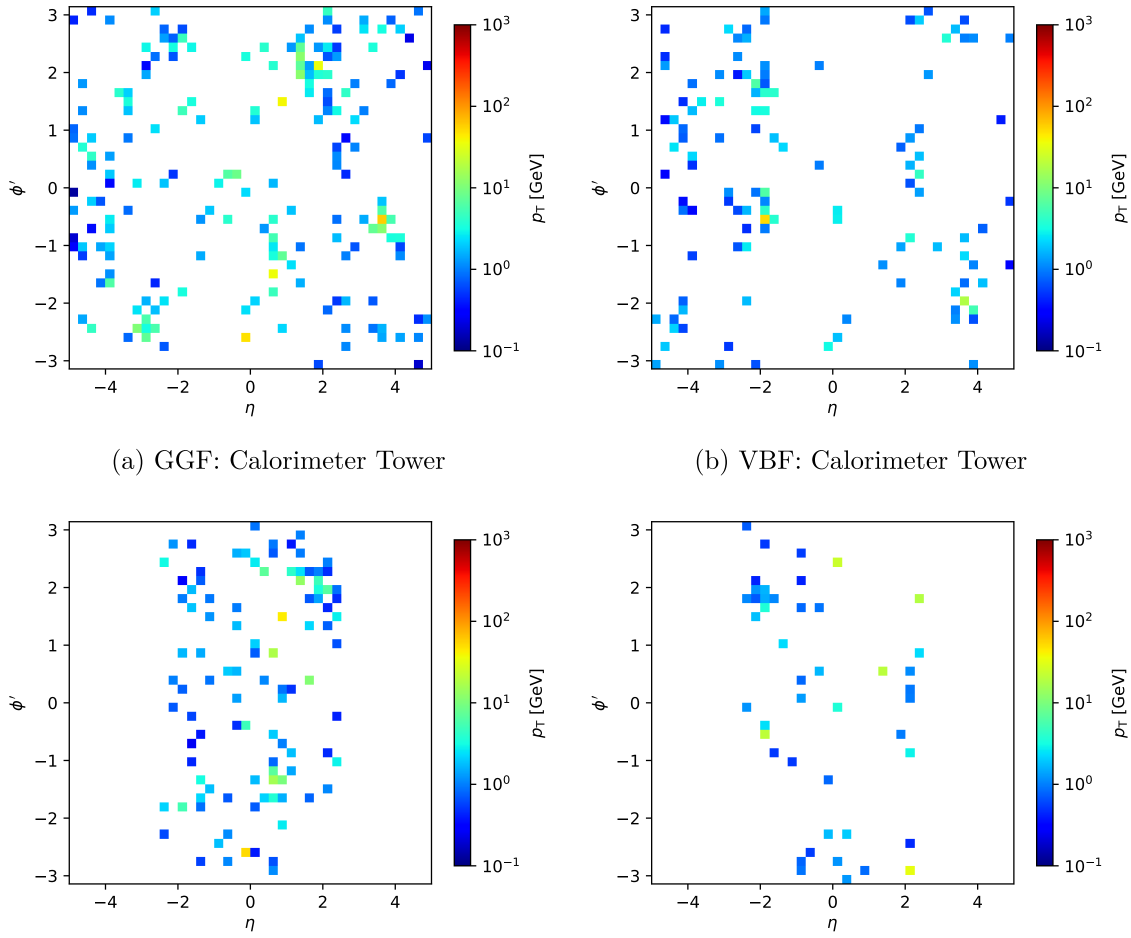
13.2 Event image of $H \to ZZ^* \to 4\ell$ mode

We follow the same procedure described in section 2.3 to construct the event image. Each event image contains only two channels: calorimeter towers and tracks. To enable transferability of the trained CNN model across different decay modes, we remove decay product (lepton) information from the event images.

Figure 5 shows representative event images for the $H \to ZZ^* \to 4\ell$ mode, separated by production mechanism (GGF and VBF) and by feature type (calorimeter tower or track).

13.3 Testing results of di-photon classifier

We evaluate the performance of the $H \to \gamma\gamma$ classifier trained in section 10 on a different decay mode: $H \to ZZ^* \to 4\ell$. We focus on Case 1, where both the photon channel and



(a) GGF: Calorimeter Tower

(b) VBF: Calorimeter Tower

(c) GGF: Track

(d) VBF: Track

Figure 5: Event images for $H \to ZZ^* \to 4\ell$ events produced via GGF and VBF. Images are shown separately for the calorimeter tower and track channels.

photon-related features in the tower channel are removed from the input. This setting is designed to test whether a model trained without explicit decay production information can still extract meaningful patterns applicable to other decay modes.

The evaluation is performed using the $2q0g$ dataset for both decay channels. Table 32 summarizes the results. These results indicate that, although the classifier performs well on its original $H \to \gamma\gamma$ dataset, its performance degrades when applied to the $H \to ZZ^* \to 4\ell$ events. This suggests that even after removing photon-related features, the learned representation may still carry decay-mode-specific biases that affect cross-channel transferability.

Table 32: CNN training results for Case 1: both the photon channel and Tower photon-related features are removed. The classifier is trained on $H \to \gamma\gamma$ and evaluated on both $H \to \gamma\gamma$ and $H \to ZZ^* \to 4\ell$. The ACC and AUC are evaluated based on 10 training.

	$H \to \gamma\gamma$		$H \to ZZ^* \to 4\ell$	
Datasets	ACC	AUC	ACC	AUC
Original	0.690 ± 0.005	0.750 ± 0.008	0.621 ± 0.022	0.665 ± 0.027
+5	0.695 ± 0.008	0.747 ± 0.005	0.592 ± 0.012	0.624 ± 0.016
+10	0.698 ± 0.008	0.752 ± 0.010	0.593 ± 0.013	0.627 ± 0.020
+15	0.700 ± 0.010	0.755 ± 0.010	0.589 ± 0.012	0.621 ± 0.019

13.4 Testing results of supervised classifier

We train supervised CNN models separately on two Higgs decay channels. The training, validation, and testing dataset sizes are identical to those listed in table 11. Only events that pass all selection requirements are used. Importantly, all input features directly associated with the decay products are removed.

Table 33 summarizes the classification results. Each model is trained on one decay mode and evaluated on both $H \to \gamma\gamma$ and $H \to ZZ^* \to 4\ell$ datasets. Compared to the results in table 12, the di-photon classifier shows slightly worse performance due to the exclusion of photon-specific information.

Although both models are trained without decay product features, the CNN still captures decay-mode-specific differences. Moreover, the model trained on $H \to ZZ^* \to 4\ell$ generalizes to the $H \to \gamma\gamma$ dataset worse than the reverse case.

Table 33: CNN classification results with decay product information removed. Each model is trained on one decay channel and tested on both $H \to \gamma\gamma$ and $H \to ZZ^* \to 4\ell$. Results are averaged over 10 training runs.

	$H \to \gamma\gamma$		$H \to ZZ^* \to 4\ell$	
Training channel	ACC	AUC	ACC	AUC
$H \to \gamma\gamma$	0.775 ± 0.001	0.852 ± 0.001	0.752 ± 0.002	0.827 ± 0.003
$H \to ZZ^* \to 4\ell$	0.738 ± 0.002	0.806 ± 0.003	0.793 ± 0.001	0.872 ± 0.001

14 Larger training dataset

In section 10, we observed that removing photon information improved the training performance. There are several possible reasons for this:

- The photon channel contains large “blank” regions, which may confuse the neural network.
- Photons have higher transverse momentum, leading the network to focus on them and delay learning the jet features.

Increasing the size of the training dataset may mitigate these effects, allowing the network to learn both photon and jet patterns effectively.

We evaluate model performance using the $2q0g$ dataset at different integrated luminosities, while retaining the photon information during training. The results are summarized in table 34. These results indicate that increasing the dataset size leads to performance improvement, even when photon information is retained.

Table 34: CNN training results with different dataset sizes corresponding to various luminosities. Each model is trained with photon information included. The ACC and AUC are evaluated over 10 training runs.

	M_1/M_2		S/B	
Luminosity (fb^{-1})	ACC	AUC	ACC	AUC
3000	0.615 ± 0.005	0.632 ± 0.007	0.650 ± 0.011	0.703 ± 0.015
12000	0.643 ± 0.002	0.684 ± 0.003	0.689 ± 0.009	0.743 ± 0.011

14.1 Logarithmic transverse momentum

To confirm whether the high p_{T} of photons affects training performance, we apply a logarithmic transformation to the p_{T} values in each pixel. This aims to reduce the intensity difference between high- and low-p_{T} pixels, potentially allowing the network to better focus on spatial patterns. Specifically, each pixel's input is replaced with

$$\log\left(p_{\mathrm{T}} + 1\right) \tag{3}$$

where p_{T} is the original transverse momentum value. The addition of 1 takes care of $p_{\mathrm{T}} = 0$.

We evaluate model performance on the $2q0g$ dataset with and without the logarithmic transformation. Table 35 summarizes the results. The results indicate that applying the $\log(p_{\mathrm{T}})$ transformation does not improve the performance. This suggests that intensity dominance due to high-p_{T} photons may not be the main limiting factor, or that the network is already sufficiently robust to handle such variations.

Table 35: CNN training results with and without the logarithmic p_{T} transformation. Each model is trained with photon information included. ACC and AUC values are averaged over 10 training runs.

	M_1/M_2		S/B	
Dataset	ACC	AUC	ACC	AUC
Original	0.615 ± 0.005	0.632 ± 0.007	0.650 ± 0.011	0.703 ± 0.015
$\log(p_{\mathrm{T}})$	0.615 ± 0.006	0.635 ± 0.009	0.644 ± 0.010	0.694 ± 0.014

15 Event-CNN

We test an alternative network structure known as the event-CNN, as proposed in Ref. [9].

15.1 Supervised training with event-CNN

We apply the event-CNN architecture to supervised classification tasks. The training, validation, and testing dataset sizes are identical to those listed in table 11. Only events passing all selection criteria are used in training.

Table 36 summarizes the classification performance. Compared to the previous CNN results shown in table 12, the event-CNN achieves slightly better performance.

Table 36: Event-CNN classification results. Results are averaged over 10 training runs.

ACC	AUC
0.792 ± 0.001	0.871 ± 0.001

15.2 CWoLa training with event-CNN

We consider the di-photon case mentioned in section 7. We also apply the ϕ-shifting augmentation technique introduced in section 9.2.

Table 37 summarizes the training results using the $2q0g$ dataset with ϕ-shifting augmentation. Although ϕ-shifting improves model performance on mixed datasets, the enhancement saturates as the dataset size increases. Notably, on pure samples, the performance remains similar regardless of augmentation, indicating that the gain is primarily due to over-fitting on mixed datasets.

Table 37: CNN training results with different augmentation sizes. The ACC and AUC are evaluated based on 10 training.

	M_1/M_2		S/B	
Datasets	ACC	AUC	ACC	AUC
Original	0.624 ± 0.011	0.644 ± 0.021	0.679 ± 0.009	0.739 ± 0.012
+5	0.649 ± 0.004	0.691 ± 0.005	0.681 ± 0.005	0.739 ± 0.005
+10	0.650 ± 0.004	0.692 ± 0.004	0.678 ± 0.007	0.735 ± 0.010
+15	0.651 ± 0.004	0.694 ± 0.005	0.676 ± 0.006	0.732 ± 0.007

15.3 Remove photon information with event-CNN

To investigate the role of photon inputs in training, we apply the event-CNN model to the Case 1 datasets described in section 10.

We use the $2q0g$ datasets for evaluation. The training results are summarized in table 38. The training performance is almost the same, suggesting that the model can still extract discriminative features from the jet pattern alone.

15.4 Various training size with event-CNN

We also study the effect of dataset size on the model's performance. The event-CNN is trained using the $2q0g$ datasets corresponding to different integrated luminosities, while keeping photon information included. The results are summarized in table 39. The perfor-

Table 38: Event-CNN training results with both the photon channel and Tower photon features removed (Case 1). The ACC and AUC are evaluated based on 10 training.

	M_1/M_2		S/B	
Datasets	ACC	AUC	ACC	AUC
Original	0.624 ± 0.011	0.644 ± 0.021	0.679 ± 0.009	0.739 ± 0.012
Case 1	0.637 ± 0.012	0.667 ± 0.020	0.682 ± 0.004	0.740 ± 0.005

Table 39: Event-CNN training results with different dataset sizes corresponding to various luminosities. Each model is trained with photon information included. The ACC and AUC are evaluated over 10 training runs.

	M_1/M_2		S/B	
Luminosity (fb^{-1})	ACC	AUC	ACC	AUC
300	0.610 ± 0.015	0.589 ± 0.036	0.606 ± 0.033	0.641 ± 0.045
900	0.625 ± 0.007	0.628 ± 0.009	0.660 ± 0.014	0.715 ± 0.018
1800	0.625 ± 0.010	0.644 ± 0.019	0.674 ± 0.009	0.732 ± 0.012
3000	0.624 ± 0.011	0.644 ± 0.021	0.679 ± 0.009	0.739 ± 0.012
12000	0.655 ± 0.002	0.701 ± 0.003	0.683 ± 0.004	0.742 ± 0.005

mance improved with increased dataset size. For the pure datasets, the training performance is saturated after 3000 fb^{-1}.

We repeat the analysis using datasets where photon information has been removed. The comparison between the two scenarios is shown in figure 6. In the high-luminosity regime, the model performance is similar regardless of whether photon information is included. However, at lower luminosities, datasets without photon information unexpectedly yield better AUC.

15.5 Photon information removed datasets with ϕ-shifting augmentation

We test the ϕ-shifting augmentation of the photon information removed case. Here, we consider the luminosity of $\mathcal{L} = 100\ \text{fb}^{-1}$.

Table 40 summarizes the training results using the $2q0g$ dataset with ϕ-shifting augmentation. The results show that ϕ-shifting augmentation improves model performance. The ϕ-shifting augmentation helps mitigate statistical fluctuations and improve generalization.

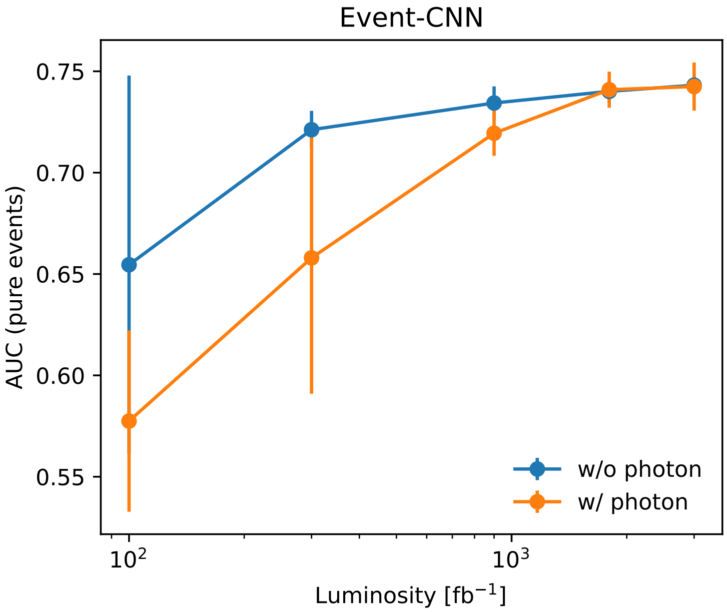


Figure 6: AUC comparison for Event-CNN models trained with and without photon information across different luminosities. The average and standard deviation are evaluated over 10 training runs.

Table 40: Event-CNN training results for the photon-removed dataset with different ϕ-shifting augmentation sizes. ACC and AUC are averaged over 10 training runs.

	M_1/M_2		S/B	
Datasets	ACC	AUC	ACC	AUC
Original	0.627 ± 0.028	0.601 ± 0.066	0.617 ± 0.064	0.655 ± 0.093
+5	0.623 ± 0.017	0.621 ± 0.036	0.657 ± 0.019	0.709 ± 0.024
+10	0.627 ± 0.018	0.634 ± 0.033	0.663 ± 0.011	0.718 ± 0.015
+15	0.627 ± 0.016	0.635 ± 0.034	0.660 ± 0.009	0.713 ± 0.011

15.6 $H \to ZZ^* \to 4\ell$ channel

We follow the same setup as described in section 13. For CWoLa training, we consider multiple luminosity scenarios. In the case of $\mathcal{L} = 3000$ fb^{-1}, the number of events in the mixed datasets is shown in table 31.

The event-CNN training results are summarized in table 41. At $\mathcal{L} = 3000$ fb^{-1}, the ACC and AUC are close to 0.5, indicating that the training is nearly ineffective. However, as the training dataset size increases to $\mathcal{L} = 30000$ fb^{-1}, the AUC for the pure dataset improves significantly, reaching values around 0.8. This highlights the importance of dataset size.

Table 41: Event-CNN training results using datasets corresponding to different luminosities. The $ZZ \to 4\ell$ decay channel is considered, with information on the decay products included. ACC and AUC are averaged over 10 training runs.

	M_1/M_2		S/B	
Luminosity (fb^{-1})	ACC	AUC	ACC	AUC
3000	0.575 ± 0.012	0.508 ± 0.039	0.538 ± 0.026	0.543 ± 0.035
30000	0.601 ± 0.010	0.621 ± 0.012	0.732 ± 0.018	0.800 ± 0.023

To assess the contribution of decay products to model performance, we train the event-CNN on the $\mathcal{L} = 30000$ fb^{-1} dataset with all lepton-related information removed. Since lepton signals can appear in both Tower and Track channels, we mask the corresponding pixels by setting them to zero in both inputs. The results are shown in table 42. The performance is 10% worse for pure events when lepton information is removed, indicating that some of the discriminative features are removed.

Table 42: Event-CNN training results after removing all decay product information. ACC and AUC are averaged over 10 training runs.

	M_1/M_2		S/B	
Datasets	ACC	AUC	ACC	AUC
Original	0.601 ± 0.010	0.621 ± 0.012	0.732 ± 0.018	0.800 ± 0.023
w/o lepton	0.606 ± 0.010	0.626 ± 0.012	0.645 ± 0.006	0.697 ± 0.009

We apply the event-CNN model trained on the photon-removed $H \to \gamma\gamma$ dataset (from section 15.5) to $H \to ZZ^* \to 4\ell$ events. The evaluation uses the $2q0g$ datasets for both decay channels, and the results are summarized in table 43. These results demonstrate the

Table 43: Event-CNN training results for phton-removed datasets. The classifier is trained on $H \to \gamma\gamma$ and evaluated on both $H \to \gamma\gamma$ and $H \to ZZ^* \to 4\ell$. The ACC and AUC are evaluated based on 10 training.

	$H \to \gamma\gamma$		$H \to ZZ^* \to 4\ell$	
Datasets	ACC	AUC	ACC	AUC
Original	0.617 ± 0.061	0.655 ± 0.089	0.599 ± 0.041	0.634 ± 0.056
+5	0.663 ± 0.015	0.717 ± 0.020	0.634 ± 0.015	0.677 ± 0.018
+10	0.667 ± 0.010	0.723 ± 0.014	0.638 ± 0.010	0.683 ± 0.011
+15	0.664 ± 0.009	0.719 ± 0.010	0.634 ± 0.006	0.679 ± 0.009

transferability of the event-CNN model across decay channels. The AUC for $ZZ \to 4\ell$ is comparable to that achieved by training directly on $ZZ \to 4\ell$ events at $\mathcal{L} = 30000$ fb^{-1}.

This suggests that, by leveraging both dataset augmentation and transfer learning, we can effectively simulate the performance equivalent to having approximately 29900 fb^{-1} of additional data. This approach offers a practical path to improving tagger performance in channels with limited statistics.

15.7 Supervised learning without photon information

We perform the supervised training of the event-CNN without photon information to explore the importance of the photon-related features.

Table 44 and figure 7 summarize the training results. The event-CNN achieves slightly better performance when photon information is included. However, the small difference in AUC indicates that the model is still able to extract meaningful features from the jet pattern.

Table 44: Event-CNN classification results. Results are averaged over 10 training runs. The 10 training runs use the same dataset, only the initial weights are different.

Datasets	ACC	AUC
w/ photon	0.792 ± 0.001	0.871 ± 0.001
w/o photon	0.784 ± 0.001	0.862 ± 0.001

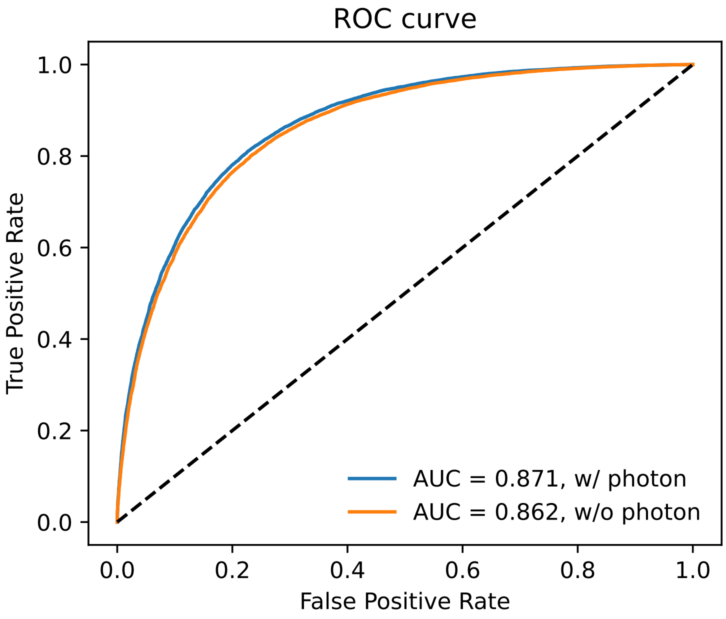


Figure 7: ROC curves for event-CNN models trained with and without photon information.

16 Particle transformer

We test an alternative network structure known as the Particle Transformer (ParT), as proposed in Ref. [10].

16.1 Supervised training with ParT

We apply the ParT architecture to supervised classification tasks. The training, validation, and testing dataset sizes are identical to those listed in table 11. Only events passing all selection criteria are used in training.

Table 45 summarizes the training results. Compared to the previous results shown in table 44 for event-CNN architecture, the ParT achieves slightly worse performance.

Table 45: ParT classification results. Results are averaged over 10 training runs. The 10 training runs use the same dataset, only the initial weights are different.

Datasets	ACC	AUC
w/ photon	0.785 ± 0.013	0.862 ± 0.014
w/o photon	0.773 ± 0.019	0.850 ± 0.022

16.2 CWoLa training with ParT

We also study the effect of dataset size on the model's performance. The ParT is trained using the $2q0g$ datasets corresponding to different integrated luminosities. Table 46 summarizes the training results with and without photon information.

Table 46: ParT training results with different dataset sizes corresponding to various luminosities. The ACC and AUC are evaluated over 10 training runs.

(a) With photon

	M_1/M_2		S/B	
Luminosity (fb^{-1})	ACC	AUC	ACC	AUC
100	0.603 ± 0.015	0.539 ± 0.041	0.544 ± 0.043	0.540 ± 0.078
300	0.607 ± 0.015	0.549 ± 0.066	0.572 ± 0.071	0.584 ± 0.112
900	0.620 ± 0.013	0.617 ± 0.060	0.657 ± 0.062	0.705 ± 0.099
1800	0.609 ± 0.018	0.575 ± 0.071	0.607 ± 0.086	0.638 ± 0.117
3000	0.622 ± 0.019	0.607 ± 0.088	0.648 ± 0.097	0.681 ± 0.152

(b) Without photon

	M_1/M_2		S/B	
Luminosity (fb^{-1})	ACC	AUC	ACC	AUC
100	0.612 ± 0.019	0.544 ± 0.061	0.545 ± 0.044	0.536 ± 0.082
300	0.602 ± 0.012	0.535 ± 0.062	0.556 ± 0.046	0.564 ± 0.072
900	0.610 ± 0.020	0.576 ± 0.064	0.595 ± 0.080	0.625 ± 0.110
1800	0.620 ± 0.022	0.592 ± 0.098	0.637 ± 0.105	0.661 ± 0.168
3000	0.637 ± 0.008	0.667 ± 0.015	0.714 ± 0.022	0.782 ± 0.026

The comparison between the two scenarios is shown in figure 8. The training performance is very unstable in both cases. Some trainings have totally failed.

16.3 ϕ-shifting augmentation on ParT

We test the ϕ-shifting augmentation of the photon information removed case. Here, we consider the luminosity of $\mathcal{L} = 100\ \text{fb}^{-1}$.

Table 47 summarizes the training results using the $2q0g$ dataset with ϕ-shifting augmentation. The results show that ϕ-shifting augmentation improves model performance. The ϕ-shifting augmentation helps improve generalization, but can not mitigate the statistical fluctuation.

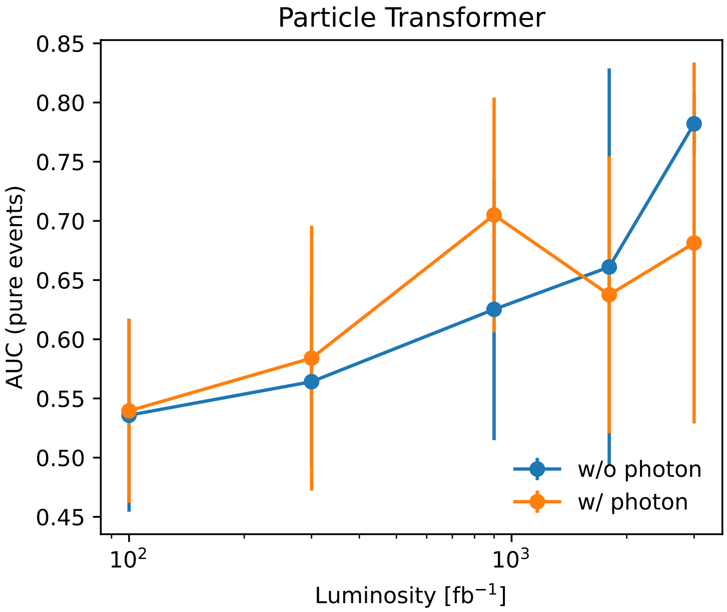


Figure 8: AUC comparison for ParT models trained with and without photon information across different luminosities. The average and standard deviation are evaluated over 10 training runs.

Table 47: ParT training results for the photon-removed dataset with different ϕ-shifting augmentation sizes. ACC and AUC are averaged over 10 training runs.

	M_1/M_2		S/B	
Datasets	ACC	AUC	ACC	AUC
Original	0.612 ± 0.019	0.544 ± 0.061	0.545 ± 0.044	0.536 ± 0.082
+5	0.595 ± 0.005	0.495 ± 0.031	0.515 ± 0.014	0.492 ± 0.042
+10	0.606 ± 0.017	0.552 ± 0.065	0.579 ± 0.063	0.599 ± 0.094
+15	0.630 ± 0.028	0.620 ± 0.065	0.646 ± 0.066	0.693 ± 0.093

16.4 Implementing p_T normalization with ParT

We apply p_T normalization to investigate whether it can mitigate training fluctuations. The ParT model is trained on the $2q0g$ datasets corresponding to different integrated luminosities.

The comparison between the two setups is shown in figure 9. In both cases, the training performance remains unstable, and in some runs, the training completely fails.

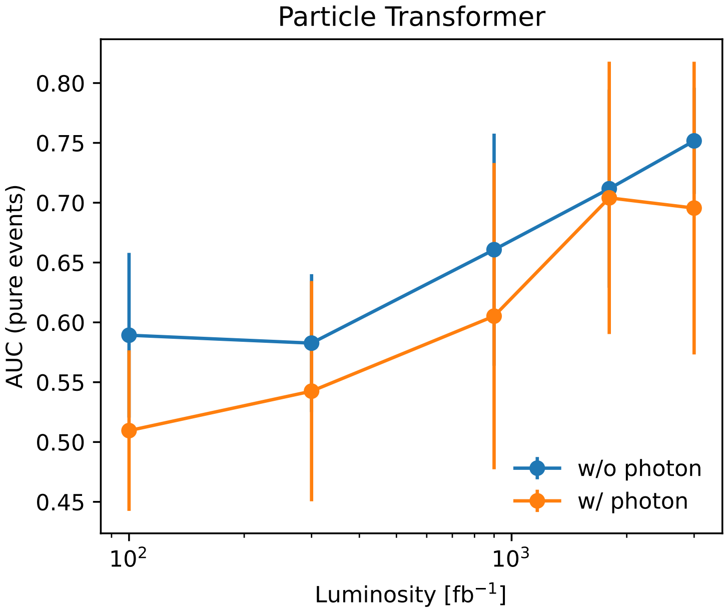


Figure 9: AUC comparison for ParT models trained with p_T normalization across different luminosities. The average and standard deviation are evaluated over 10 independent training runs.

16.5 Implementing a learning rate schedule

We introduce a learning rate schedule with a warm-up to prevent the training from diverging in the early stage. The schedule combines a short warm-up phase with a subsequent exponential decay strategy. The ParT model is trained on the $2q0g$ datasets corresponding to different integrated luminosities.

The comparison between the two scenarios is shown in figure 10. The training performance still exhibits instabilities, but the completely failed cases are fewer than in previous testing.

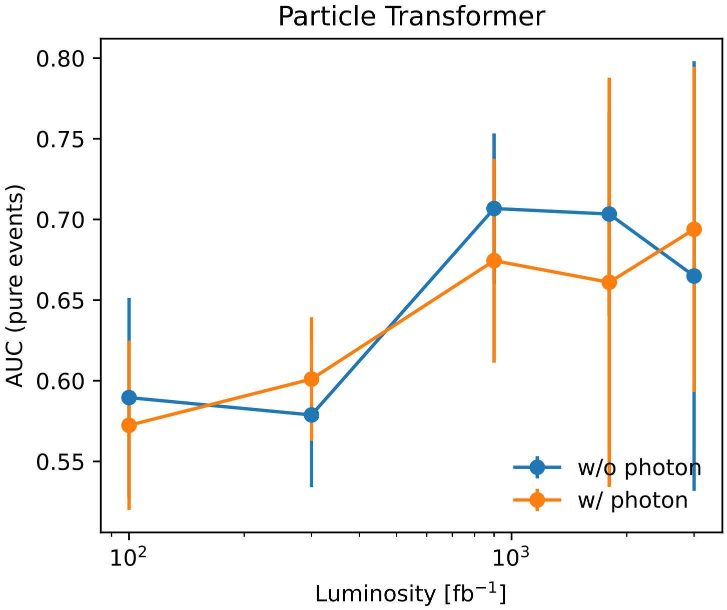


Figure 10: AUC comparison for ParT models trained with learning rate schedule across different luminosities. The average and standard deviation are evaluated over 10 independent training runs.

17 Comparison of training sample

There are some differences between Yian's and my training sample preparation workflows:

1. Yian did not apply flipping.
2. For the ϕ-translation, I computed a common ϕ-center for all channels, while Yian computed the ϕ-center for each channel separately.

It is not immediately clear how these differences affect the training results. To investigate this, I tested supervised and CWoLa training on event-CNN using Yian's samples.

For the supervised training, we used the same setup as in section 15.1. Table 48 summarizes the classification performance. Yian's results are worse than FY's by about 1% for the photon case.

For CWoLa training, we trained models with different luminosities. Table 49 summarizes the results with and without photon information.

The comparison among different cases is shown in figure 11. Overall, Yian's results are worse than FY's. These results suggest that differences in the sample preparation workflow have a direct impact on the training performance.

Table 48: Event-CNN classification results using Yian's and FY's samples. Results are averaged over 10 training runs.

Datasets	Yian		FY	
	ACC	AUC	ACC	AUC
w/ photon	0.782 ± 0.001	0.859 ± 0.001	0.792 ± 0.001	0.871 ± 0.001
w/o photon	0.781 ± 0.001	0.858 ± 0.001	0.784 ± 0.001	0.862 ± 0.001

Table 49: Event-CNN training results using Yian's samples with different dataset sizes corresponding to various luminosities. Results are averaged over 10 training runs.

(a) With photon

	M_1/M_2		S/B	
Luminosity (fb^{-1})	ACC	AUC	ACC	AUC
100	0.604 ± 0.017	0.517 ± 0.054	0.560 ± 0.046	0.572 ± 0.074
300	0.603 ± 0.009	0.585 ± 0.021	0.620 ± 0.006	0.662 ± 0.009
900	0.609 ± 0.006	0.606 ± 0.011	0.640 ± 0.005	0.694 ± 0.007
1800	0.609 ± 0.006	0.616 ± 0.013	0.652 ± 0.011	0.708 ± 0.015
3000	0.615 ± 0.005	0.630 ± 0.012	0.665 ± 0.015	0.724 ± 0.018

(b) Without photon

	M_1/M_2		S/B	
Luminosity (fb^{-1})	ACC	AUC	ACC	AUC
100	0.608 ± 0.023	0.541 ± 0.069	0.553 ± 0.039	0.563 ± 0.061
300	0.607 ± 0.009	0.590 ± 0.026	0.610 ± 0.024	0.651 ± 0.034
900	0.611 ± 0.005	0.612 ± 0.012	0.637 ± 0.009	0.691 ± 0.011
1800	0.611 ± 0.007	0.619 ± 0.013	0.656 ± 0.017	0.712 ± 0.021
3000	0.620 ± 0.009	0.640 ± 0.018	0.669 ± 0.016	0.728 ± 0.020

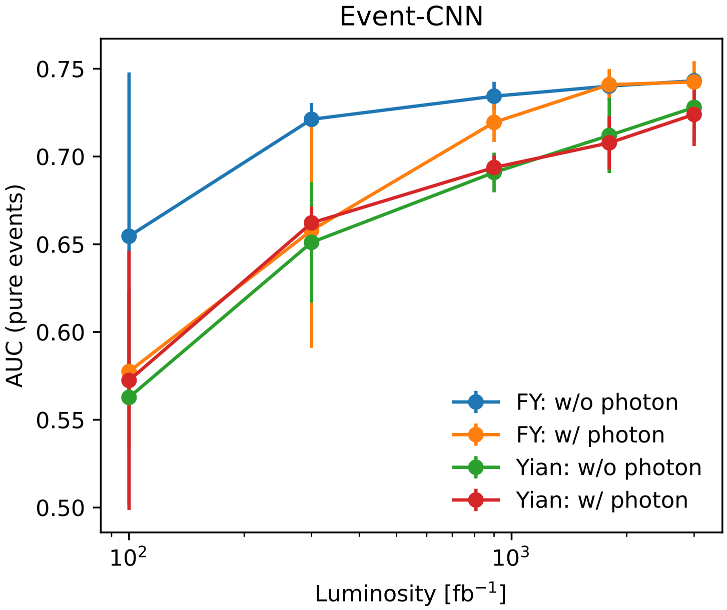


Figure 11: AUC comparison for event-CNN models trained with and without photon information across different luminosities. Results are averaged over 10 training runs, with error bars indicating standard deviations.

The following step is included for Yian's pre-processing flow:

- another moving in the ϕ-direction. This moves to ensure the high intensity p_T region does not cross the ±π boundary.

The comparison among different cases is shown in figure 12. Overall, Yian's results are consistent than FY's. The main differences are the without photon case in low luminosities.

18 Modify the training setup for Transformer

The training results of the Transformer model are highly unstable in my setup, a behavior not observed in Yian's training. To investigate the possible cause, we compared the pre-processed datasets. The inputs for the with and without photon cases yield almost identical values after preprocessing, suggesting that the instability is not due to discrepancies in the input samples.

In the comparison, we unexpectedly found that roughly 0.1% VBF events contain no constituents in the TRACK channel. Since the Particle Transformer can handle NaN values, we do not expect this issue to be the primary cause of the training failures.

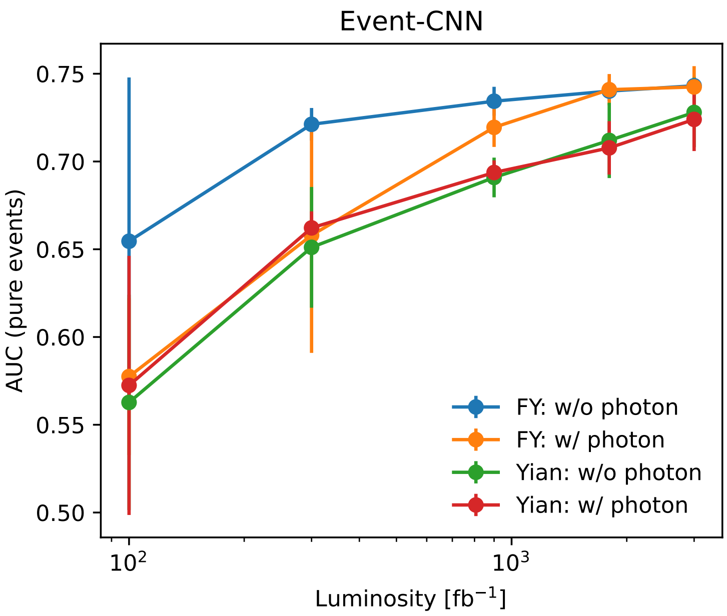


Figure 12: AUC comparison for event-CNN models trained with and without photon information across different luminosities. Results are averaged over 10 training runs, with error bars indicating standard deviations.

We further tested a range of hyperparameter configurations, varying the batch size from 128 to 512 and the learning rate from 10^{-5} to 4×10^{-4}. However, all of these setups still resulted in unstable training, with the runs failing.

Finally, we found that this issue stems from the "logit output." The logit refers to the raw classifier output before applying a normalization function such as sigmoid or softmax. In my original setup, the loss function did not correctly account for this, leading to inconsistent optimization behavior. The recommended usage is to specify `from_logits=True` in the `BinaryCrossentropy` function, as shown below:

```
model.compile(
    loss=keras.losses.BinaryCrossentropy(from_logits=True),
    ...
)
```

After fixing this issue, the training process became much more stable. Figure 13 presents the updated results. In both the with- and without-photon cases, the models now exhibit consistent training behavior, and no completely failed runs are observed.

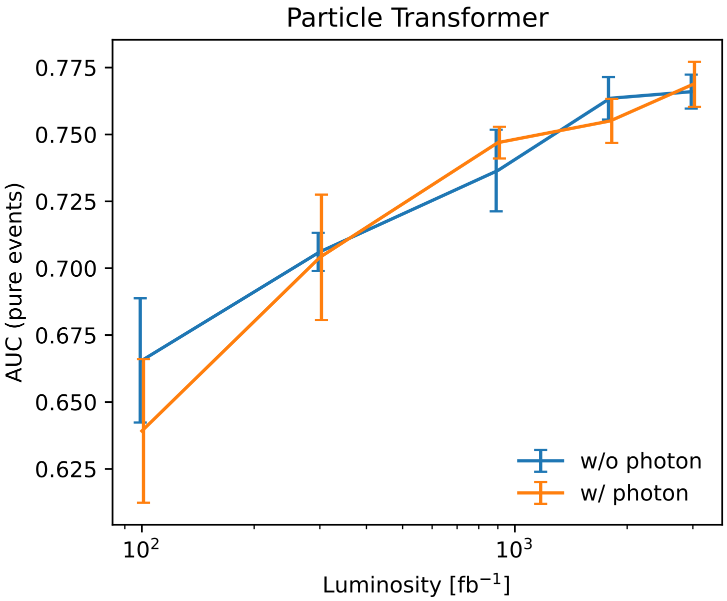


Figure 13: AUC comparison for ParT models trained with p_{T} normalization across different luminosities. Results are averaged over 10 independent training runs, with error bars indicating standard deviations.

19 Summary of all tests

19.1 Supervised training

Table 50 summarizes the supervised training results for both Event-CNN and ParT models, with and without photon information. Each result is averaged over ten independent training runs using the same dataset but different random initializations.

Table 50: Supervised classification performance of Event-CNN and ParT models trained on the di-photon dataset. Results are averaged over 10 independent runs; only the random initialization differs between runs.

	w/ photon		w/o photon	
Model	ACC	AUC	ACC	AUC
Event-CNN	0.792 ± 0.001	0.871 ± 0.001	0.792 ± 0.001	0.871 ± 0.001
ParT	0.784 ± 0.001	0.862 ± 0.001	0.784 ± 0.001	0.862 ± 0.001

19.2 CWoLa training

Figure 14 presents the classification performance obtained using the CWoLa training strategy for both models across different integrated luminosities.

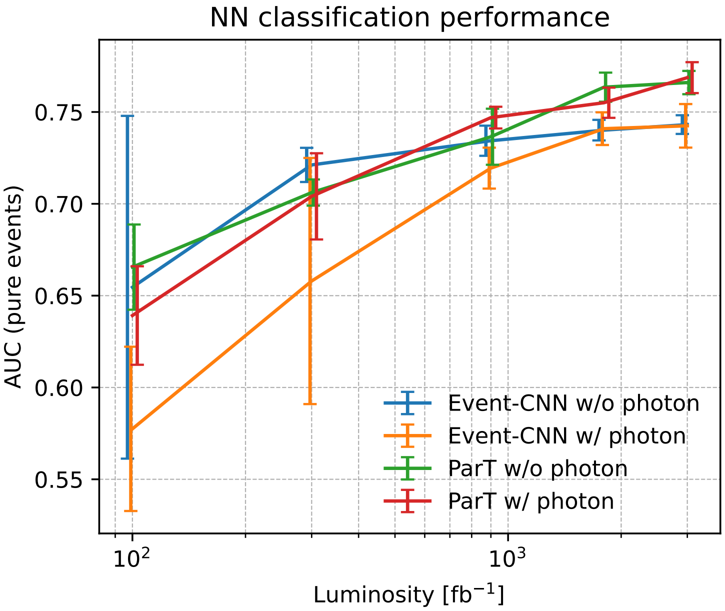


Figure 14: AUC comparison for Event-CNN and ParT models trained with the CWoLa approach across different luminosities. Results are averaged over 10 independent training runs, with error bars indicating standard deviations.

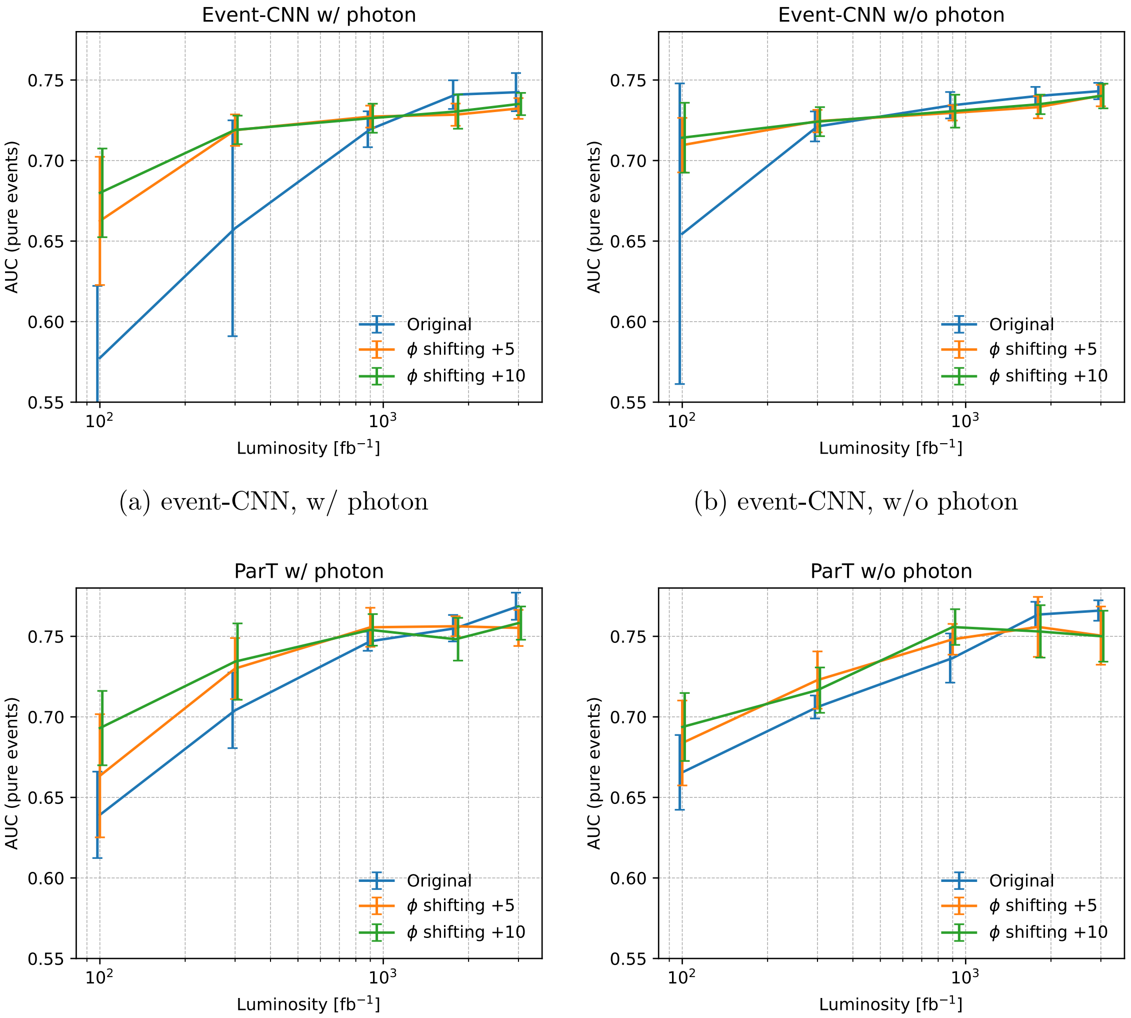
19.3 CWoLa training with data augmentation

The CWoLa training is further performed with data augmentation to enhance the training stability and sensitivity. Figure 15 shows the AUC performance across different configurations. Data augmentation improves the overall consistency and slightly enhances the average AUC, particularly in low-luminosity cases.

19.4 Application to the $H \to ZZ^* \to 4\ell$ channel

Finally, the di-photon–trained models are applied to the $H \to ZZ^* \to 4\ell$ events to test their transferability across decay modes. Their classification performance is compared with CWoLa models directly trained on the $ZZ4\ell$ dataset at very high luminosity.

Figure 16 shows the corresponding AUC comparisons. The transferred models retain



(a) event-CNN, w/ photon

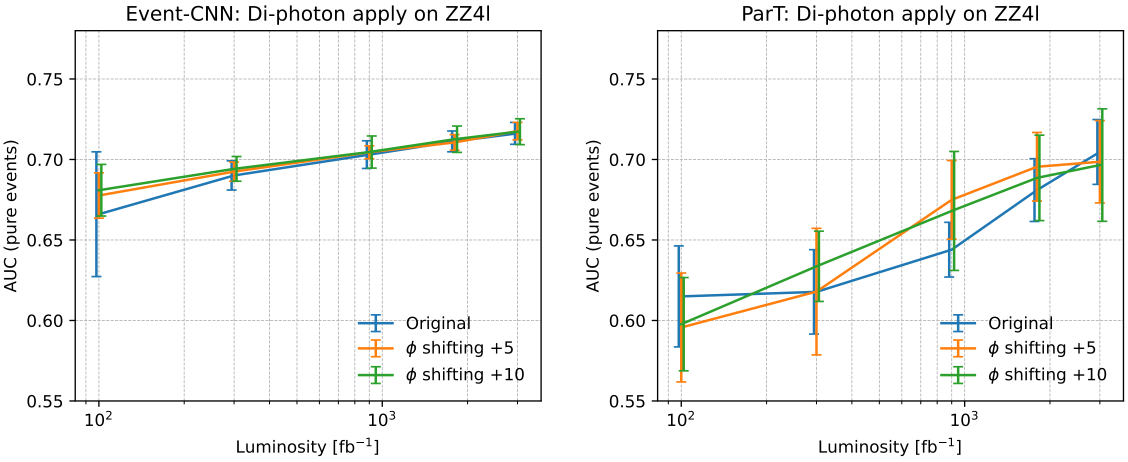
(b) event-CNN, w/o photon

(c) ParT, w/ photon

(d) ParT, w/o photon

Figure 15: AUC comparison for CWoLa training with data augmentation under various conditions. Results are averaged over 10 independent training runs.

discriminating power, suggesting that both Event-CNN and ParT can capture production-related features independent of the specific Higgs decay channel.



(a) event-CNN, w/o photon

(b) ParT, w/o photon

Figure 16: AUC comparison for di-photon−trained models applied to the $H \to ZZ^* \to 4\ell$ channel. Results are compared with CWoLa models trained directly on the $ZZ4\ell$ dataset.

References

[1] J. Alwall, R. Frederix, S. Frixione, V. Hirschi, F. Maltoni, O. Mattelaer, H. S. Shao, T. Stelzer, P. Torrielli, and M. Zaro, “The automated computation of tree-level and next-to-leading order differential cross sections, and their matching to parton shower simulations,” *JHEP*, vol. 07, p. 079, 2014.

[2] T. Sjöstrand, S. Ask, J. R. Christiansen, R. Corke, N. Desai, P. Ilten, S. Mrenna, S. Prestel, C. O. Rasmussen, and P. Z. Skands, “An introduction to PYTHIA 8.2,” *Comput. Phys. Commun.*, vol. 191, pp. 159–177, 2015.

[3] J. de Favereau, C. Delaere, P. Demin, A. Giammanco, V. Lemaître, A. Mertens, and M. Selvaggi, “DELPHES 3, A modular framework for fast simulation of a generic collider experiment,” *JHEP*, vol. 02, p. 057, 2014.

[4] M. Cacciari, G. P. Salam, and G. Soyez, “FastJet User Manual,” *Eur. Phys. J. C*, vol. 72, p. 1896, 2012.

[5] M. Cacciari, G. P. Salam, and G. Soyez, “The anti-k_t jet clustering algorithm,” *JHEP*, vol. 04, p. 063, 2008.

[6] A. Butter *et al.*, “The Machine Learning landscape of top taggers,” *SciPost Phys.*, vol. 7, p. 014, 2019.

[7] L. de Oliveira, M. Kagan, L. Mackey, B. Nachman, and A. Schwartzman, “Jet-images — deep learning edition,” *JHEP*, vol. 07, p. 069, 2016.

[8] G. Kasieczka, T. Plehn, M. Russell, and T. Schell, “Deep-learning Top Taggers or The End of QCD?,” *JHEP*, vol. 05, p. 006, 2017.

[9] C.-W. Chiang, D. Shih, and S.-F. Wei, “VBF vs. GGF Higgs with Full-Event Deep Learning: Towards a Decay-Agnostic Tagger,” *Phys. Rev. D*, vol. 107, no. 1, p. 016014, 2023.

[10] H. Qu, C. Li, and S. Qian, “Particle Transformer for Jet Tagging,” 2 2022.