Note

Feng-Yang Hsieh

1 SPANet2

Version 2 of SPANet. call it SPANet2.

1.1 Defining event topology

Defining the event topology in .ymal file. The structure of the .yaml file follows this format:

```
INPUTS:
    SEQUENTIAL:
```

1.2 Creating training dataset

.hdf5

1.3 Training options

1.4 Training

Training:

```
python -m spanet.train -of <OPTIONS_FILE> --log_dir <LOG_DIR>  --name <NAME>
```

<OPTIONS_FILE>: JSON file with options. <LOG_DIR>: output directory. <NAME>: subdirectory name

Evaluation:

```
python -m spanet.test <log_directory> -tf <TEST_FILE>
```

<log_directory>: directory containing the checkpoint and options file. <TEST_FILE> will replace the test file in the option file.

Prediction:

```
python predict.py <log_directory> <output name> -tf <TEST_FILE> --gpu ???
```

2 Test SPANet2

2.1 SM SPANet

Generate the correct format $\kappa_\lambda = 1$ training data for SPANet2 training.

- Training sample:
  - Total sample size: 76,131
  - 1h sample size: 14,527
  - 2h sample size: 60,122
  - 5% used on validation
- Testing sample:
  - Total sample size: 8,460
  - 1h sample size: 1,577
  - 2h sample size: 6,744

The training results are presented in Table 1.

Table 1: SPA-NET2 training results on the SM di-Higgs samples.

N_{Jet}	Event Fraction	Event Efficiency	Higgs Efficiency
$= 4$	0.280	0.907	0.907
$= 5$	0.287	0.806	0.847
≥ 6	0.229	0.679	0.753
Total	0.797	0.805	0.841

2.2 $\kappa 5$ SPANet

Generate the $\kappa_\lambda = 5$ training data for SPANet2 training.

- Training sample:
  - Total sample size: 78,388

  - 1h sample size: 16,013
  - 2h sample size: 59,180
  - 5% used on validation
- Testing sample:
  - Total sample size: 8,710
  - 1h sample size: 1,846
  - 2h sample size: 6,486

The training results are presented in Table 2.

Table 2: SPA-NET2 training results on the di-Higgs $\kappa_\lambda = 5$ samples.

N_{Jet}	Event Fraction	Event Efficiency	Higgs Efficiency
$= 4$	0.315	0.689	0.689
$= 5$	0.255	0.617	0.639
≥ 6	0.174	0.499	0.544
Total	0.745	0.620	0.638

2.3 Resonant SPANet

Generate the correct format resonant training data for SPANet2 training.

- Training sample:
  - Total sample size: 51,145
  - 1h sample size: 9,320
  - 2h sample size: 40,991
  - 5% used on validation
- Testing sample:
  - Total sample size: 5,683
  - 1h sample size: 1,011
  - 2h sample size: 4,582

The training results are presented in Table 3.

Table 3: SPA-NET2 training results on the resonant di-Higgs samples.

N_{Jet}	Event Fraction	Event Efficiency	Higgs Efficiency
$= 4$	0.316	0.930	0.930
$= 5$	0.282	0.808	0.839
≥ 6	0.208	0.660	0.727
Total	0.806	0.818	0.846

2.4 Mixing κ_λ SPANet

Generate the correct format mixing κ_λ training data for SPANet2 training.

- Training sample:
  - Total sample size: 51,145
  - 1h sample size: 9,320
  - 2h sample size: 40,991
  - 5% used on validation
- Testing sample:
  - Total sample size: 5,683
  - 1h sample size: 1,011
  - 2h sample size: 4,582

The training results are presented in Table 4.

Table 4: SPA-NET2 training results on the resonant di-Higgs samples.

N_{Jet}	Event Fraction	Event Efficiency	Higgs Efficiency
$= 4$	0.316	0.930	0.930
$= 5$	0.282	0.808	0.839
≥ 6	0.208	0.660	0.727
Total	0.806	0.818	0.846

2.5 Summary

In most cases, the performance of SPANet2 is worse than the old one. Some default options are different between the two versions. But even if the options are set as identical, the training results also cannot be better.

The training results of old and new versions SPANet have been summarized in Table 5.

Table 5: SPA-NET2 training results on the resonant di-Higgs samples.

	Event efficiency	
	SPANet	SPANet2
SM	0.868	0.805
kappa 5	0.725	0.620
Resonant	0.903	0.818
Mixing κ_λ	0.833	0.830

3 Combine jet assignment and event classification

This section trains the SPANet2 on the jet assignment and event classification task at the same time. This is the new feature of SPANet2.

3.1 $\kappa_\lambda = 5$ sample

For the jet assignment part, use the same sample as in Sec. 2.2.

- Training sample:
  - Total sample size: 168,125
  - Signal sample size: 78,388
  - Background sample size: 89,737
  - 5% used on validation
- Testing sample:
  - Total sample size: 18,681
  - Signal sample size: 8,710

– Background sample size: 9,971

The training results are presented in Table 6.

Table 6: SPA-NET2 training results on the $\kappa_\lambda = 5$ samples.

N_{Jet}	Event Fraction	Event Efficiency	Higgs Efficiency
$= 4$	0.316	0.930	0.930
$= 5$	0.282	0.808	0.839
≥ 6	0.208	0.660	0.727
Total	0.806	0.818	0.846

3.2 Mixing κ_λ

3.2.1 Training samples

For signal, set $\kappa_\lambda = [-5, -3, -1, 1, 2, 3, 5, 7, 9, 12]$ and generate 9,000 samples on each κ_λ point for training. The training samples are required to pass the "Four tag cut", i.e., there are at least four b-tagged jets with $p_{\text{T}} > 40$ GeV and $|\eta| < 2.5$.

Note that the κ_λ value is an input feature. For the background sample, the input κ_λ value is randomly chosen from the above values.

For the jet assignment part,

- Training sample:
  - Total sample size: 90,000
  - 1h sample size: 18,020
  - 2h sample size: 69,267
  - 5% used on validation
- Testing sample:
  - Total sample size: 9,000
  - 1h sample size: 1,802
  - 2h sample size: 6,899

For event classification,

- Training sample:

  - Total sample size: 179,737
  - Signal sample size: 90,000
  - Background sample size: 89,737
  - 5% used on validation

- Testing sample:
  - Total sample size: 18,971
  - Signal sample size: 9,000
  - Background sample size: 9,971

3.2.2 Hyperparameters setting

Some options are different between SPANet and SPANet2. List the different options below

- `hidden_dim`: 128 $\to$ 64
- `learning_rate`: 0.0007 $\to$ 0.0015
- `num_attention_heads`: 8 $\to$ 4

The total loss function consists of assignment loss and classification loss. The same weights are assigned to these losses.

- `assignment_loss_scale`: 1.0
- `classification_loss_scale`: 1.0

3.2.3 Training results

The jet assignment training results are presented in Table 7.

Table 7: SPANet2 training results on the mixing κ_λ samples.

N_{Jet}	Event Fraction	Event Efficiency	Higgs Efficiency
$= 4$	0.139	0.866	0.866
$= 5$	0.130	0.806	0.834
≥ 6	0.095	0.704	0.766
Total	0.364	0.802	0.829

Table 8 presents the classification training results.

Table 8: The SPANet2 classification training results with mixing κ_λ sample.

Training sample	ACC	AUC
Mixing κ_λ	0.828	0.911

4 κ_λ constraints on SPANet2

4.1 SPANet2 classification

Use the SPANet2 to do the signal background classification task.

When an event is put in SPANet2, SPANet2 will return a signal score p_{signal} which represents the confidence of this event is a signal event. The requirement of $p_{\text{signal}} > p_{\text{th}}$ is imposed for event selection, where $p_{\text{th}} = 0.90$. In where do not choose the value which can maximize the $S/\sqrt{B}$. Because the value is too close to 1, then very few events can pass this selection. Thus we can not do further analysis.

Set the κ_λ limits by the profile likelihood method and CLs method. Table 9 results from κ_λ constraints.

Table 9: The κ_λ constraints of SPANet2.

	Expected Constraint			
	Profile likelihood		CLs	
Selection method	Lower	Upper	Lower	Upper
SPANet2	-3.48	9.18	-3.41	9.09

5 Comparision with previous results

This section summary the results among the "min-ΔR DNN", "κ5 SPANet DNN", "mixing κ SPANet2".

5.1 Pairing performance

Figure 1 shows the pairing efficiency of different methods. Where the mixing κ SPANet2 has the best performance.

5.2 Classification performance

Table 8 presents the classification training results.

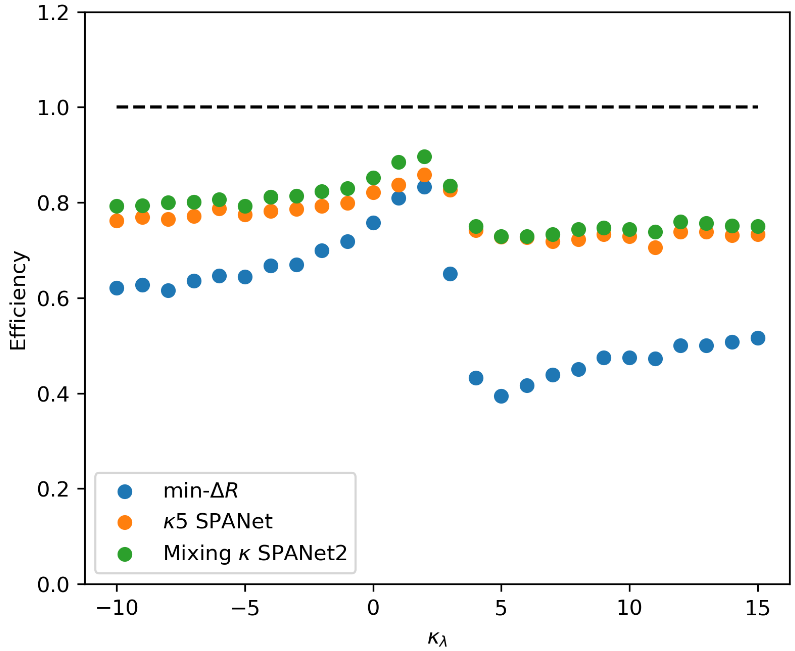


Figure 1: The pairing performance for different κ_λ samples.

Table 10: The classification performance of different selection methods.

Selection method	ACC	AUC
min-ΔR DNN	0.783	0.864
κ5 SPANet DNN	0.792	0.875
mixing κ SPANet2	0.828	0.911

5.3 κ_λ constraints

Table 16 is the κ_λ constraints of the different selection methods.

Table 11: The κ_λ constraints of different selection methods.

		Expected Constraints			
		Profile likelihood		CLs	
Pairing method	Selection method	Lower	Upper	Lower	Upper
min-ΔR	DNN	-3.81	11.16	-3.73	11.15
κ5 SPANet	DNN	-4.08	11.65	-4.02	11.68
Mixing κ SPANet2	SPANet2	-3.48	9.18	-3.41	9.09

5.4 Mass distribution plot

Figure 2 and 3 show the Higgs mass distribution for signal and background events with different pairing methods. The selection does not apply. The mass planes for the signal process all look similar for all pairing methods. For background, the results of min-ΔR are very different from others.

Figure 4 are the m_{HH} distributions after the selection.

6 SPANet2 pairing + DNN selection

This section uses the mixing κ SPANet2 for jet pairing and generates the samples for DNN training. The training samples consisted of different κ_λ value events.

6.1 Training samples

Set $\kappa_\lambda = [-5, -3, -1, 1, 2, 3, 5, 7, 9, 11]$, for each κ_λ point generate samples. For signal, the different κ_λ samples are mixed. For each type, the same number of samples is used. For background, the κ_λ values are randomly chosen from the above list. Training sample sizes are shown in Table 12.

6.2 Training results

The DNN training results are summarized in Table 13.

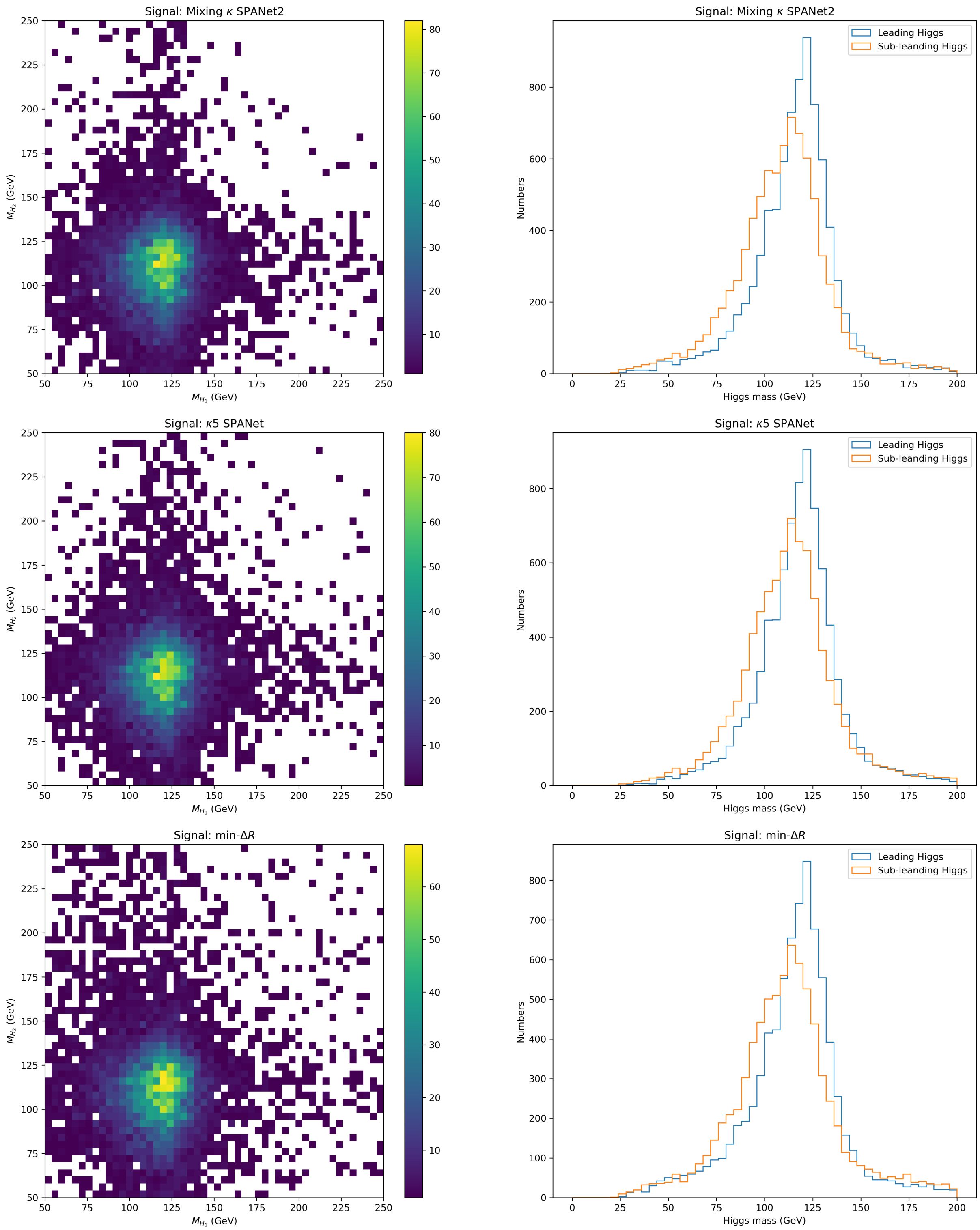


Figure 2: The mass plane and distribution of Higgs candidate for signal events with different pairing methods. The top one is mixing κ SPANet2 pairing, the middle one is $\kappa 5$ SPANet pairing, bottom one is min-ΔR pairing.

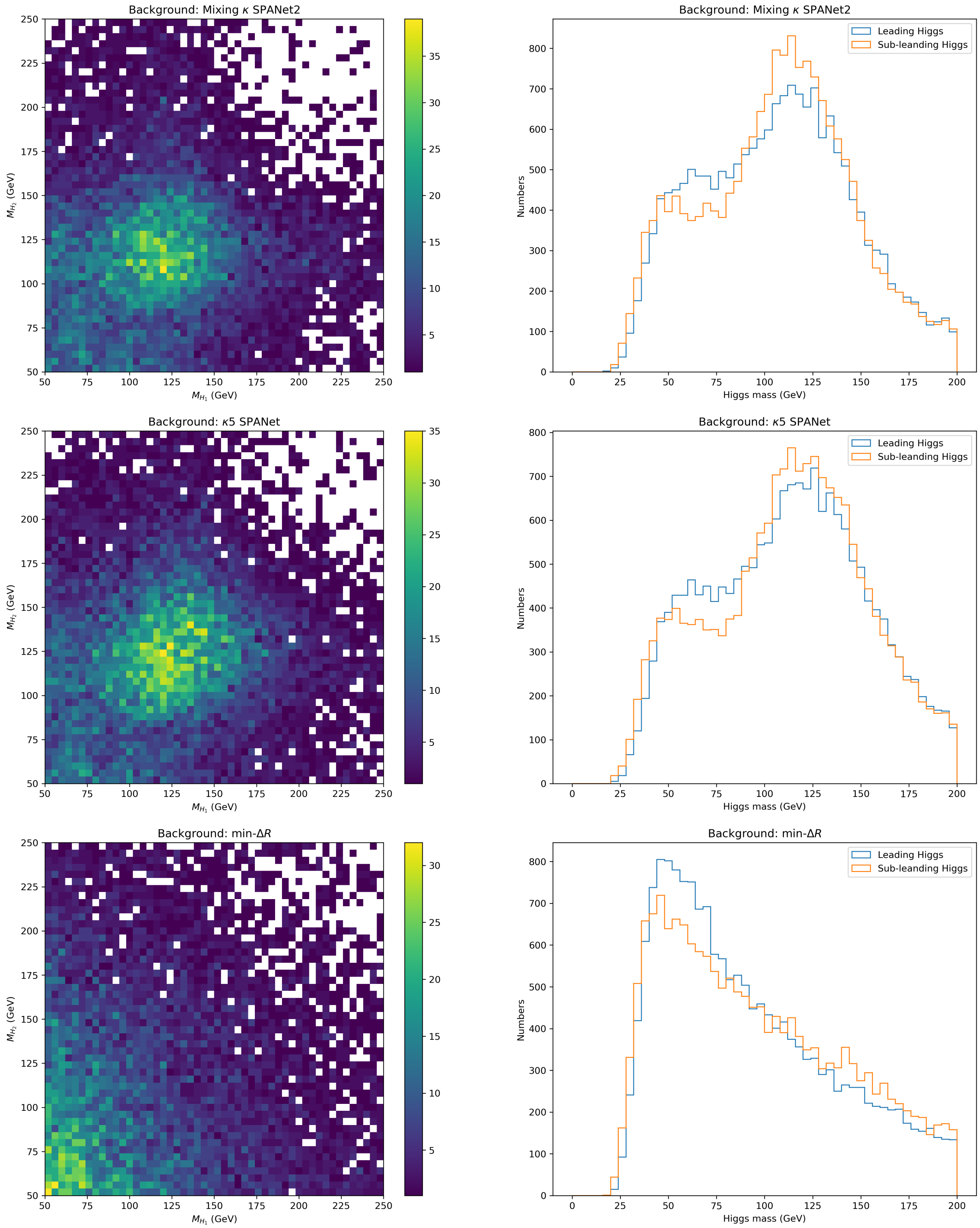
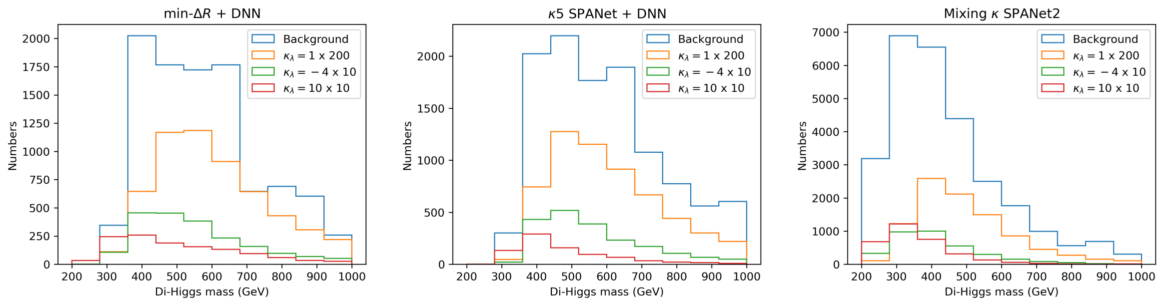


Figure 3: The mass plane and distribution of Higgs candidate for background events with different pairing methods. The top one is mixing κ SPANet2 pairing, the middle one is $\kappa 5$ SPANet pairing, bottom one is min-ΔR pairing.



(a) min-ΔR DNN (b) $\kappa 5$ SPANet DNN (c) Mixing κ SPANet2

Figure 4: The m_{HH} distributions after selection. The DNNs are trained with different pairing method samples.

Table 12: The sample size for signal and background, which are the training sample size plus the testing sample size.

Signal	Background
80k + 8k	80k + 8k

Table 13: The DNN training results with different pairing methods. The training samples contain different κ_λ samples. The average and standard deviation of 10 training are presented.

Pairing method	ACC	AUC
min-ΔR	0.800 ± 0.010	0.882 ± 0.010
$\kappa 5$ SPA-NET	0.794 ± 0.004	0.876 ± 0.004
mixing κ SPANet2	0.800 ± 0.003	0.882 ± 0.004

6.3 κ_λ limits

Set the κ_λ limits by the profile likelihood method and CLs method. The model with the best ACC is used. The p_{th} is chosen such that maximize $S/\sqrt{B}$.

The results of κ_λ constraints are summarized in Table 14.

Table 14: The κ_λ constraints with DNN selection samples.

	Expected Constraint			
	Profile likelihood		CLs	
Pairing method	Lower	Upper	Lower	Upper
min-ΔR	-3.20	10.87	-3.16	10.78
κ5 SPA-NET	-4.44	10.86	-4.35	10.78
mixing κ SPANet2	-3.42	10.85	-3.40	10.77

Here, the bug of the previous testing sample is fixed. In the previous background testing sample, all κ_λ are set to 1 not randomly chosen from κ_λ value list.

7 Summary

7.1 Classification performance

Table 15 presents the classification training results.

Table 15: The classification performance of different selection methods.

Selection method	ACC	AUC
min-ΔR DNN	0.800 ± 0.010	0.882 ± 0.010
κ5 SPANet DNN	0.794 ± 0.004	0.876 ± 0.004
mixing κ SPANet2 DNN	0.800 ± 0.003	0.882 ± 0.004
mixing κ SPANet2	0.822 ± 0.007	0.906 ± 0.006

7.2 κ_λ constraints

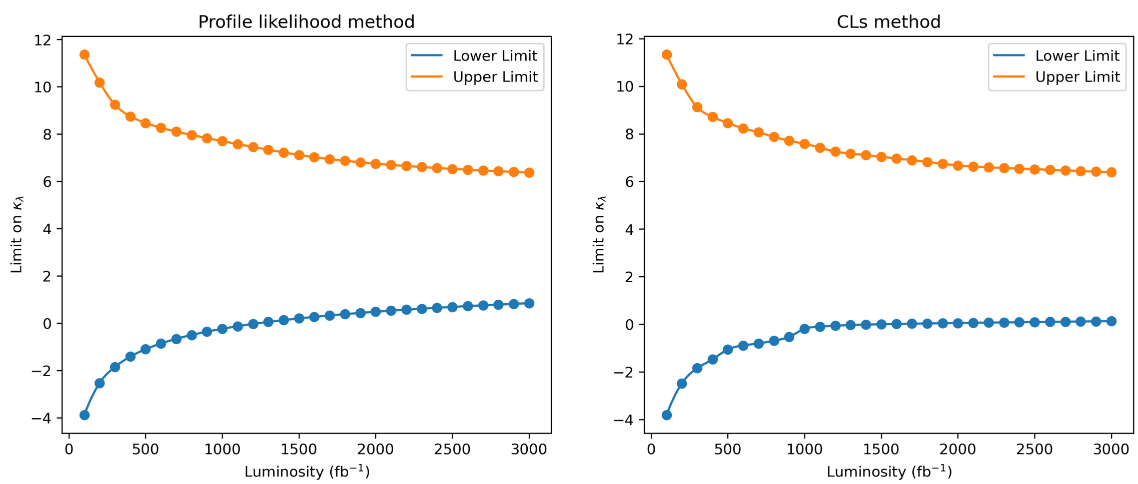
Table 16 is the κ_λ constraints of the different selection methods.

Table 16: The κ_λ constraints of different selection methods.

		Expected Constraints			
		Profile likelihood		CLs	
Pairing method	Selection method	Lower	Upper	Lower	Upper
min-ΔR	DNN	-3.20	10.87	-3.16	10.78
κ5 SPANet	DNN	-4.44	10.86	-4.35	10.78
Mixing κ SPANet2	DNN	-3.42	10.85	-3.40	10.77
Mixing κ SPANet2	SPANet2	-3.18	8.79	-3.13	8.77

8 κ_λ constraints with different luminosities

Use the min-ΔR method to set constraints with different luminosity. The results are shown in Figure 5.



(a) (b)

Figure 5: The κ_λ constraints with different luminosities. Use min-ΔR method for pairing and DNN for selection.

Use mixing κ SPANet2 for constraints setting with different luminosity. The results are presented in Table 17.

9 SPANet classifier

Figure 6 is the model structure of SPANet2. The classifier part takes the outputs of the transformer encoder. The architecture of the classifier part is just the feed-forward structure

Table 17: The κ_λ constraints with different luminosities. Use SPANet2 for event selection.

	Expected Constraints				Equivlent luminosity for min-ΔR			
	Profile likelihood		CLs		Profile likelihood		CLs	
$\mathcal{L}$ (fb^{-1})	Lower	Upper	Lower	Upper	Lower	Upper	Lower	Upper
139	-3.18	8.79	-3.13	8.77	145	388	144	381
300	-1.96	7.96	-1.96	7.89	280	810	275	796

networks.

The SPANet classifier does not take the results from the jet assignment part, because it is worse than if we just take the transformer outputs. The reason is that it can lead to worse performance due to errors in that part.

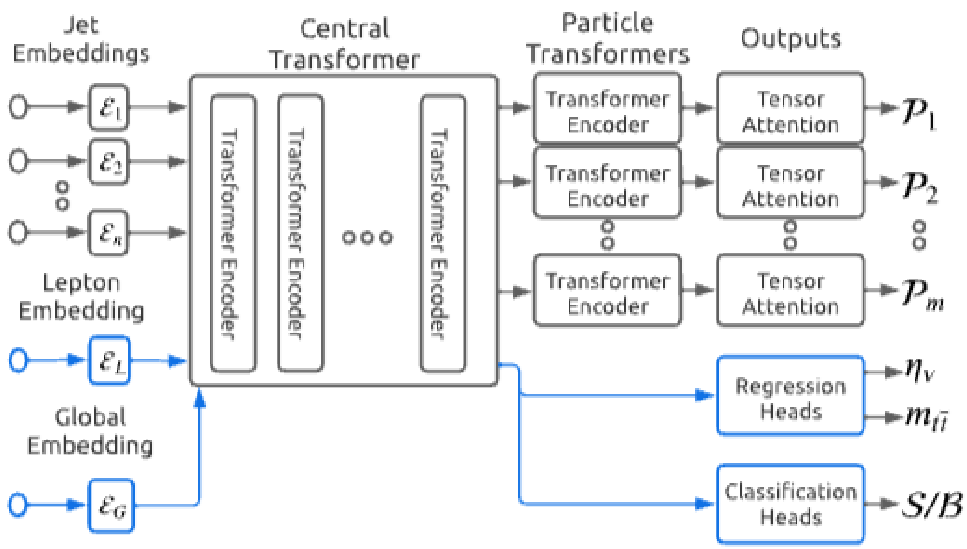


Figure 6: The model structure of SPANet2.

10 SPANet2 classification

This section turns off the jet assignment part in SPANet2 by setting the assignment loss weight to zero.

- `assignment_loss_scale`: 0.0
- `classification_loss_scale`: 1.0

Table 18: The SPANet2 classification training results with mixing κ_λ sample. The jet assignment part is turned off. The average and standard deviation of 10 training are presented.

Training sample	ACC	AUC
Mixing κ_λ	0.809 ± 0.013	0.890 ± 0.014

Using the mixing κ samples for training. The samples are the same as Sec. 3.2.1. Table 18 presents the classification training results.

Set $p_{\text{th}} = 0.95$ and use the profile likelihood method and CLs method for the κ_λ setting. Table 19 is the results of κ_λ constraints. These results are worse than simultaneously training on jet assignment and classification tasks.

Table 19: The κ_λ constraints of SPANet2.

	Expected Constraint			
	Profile likelihood		CLs	
Selection method	Lower	Upper	Lower	Upper
SPANet2	-5.07	10.98	-4.99	10.91

11 SPANet embedding vectors

The SPANet embedding vectors can be saved in `.hdf5` file by this command

```
python -m spanet.predict <log_dir> <output name> -tf <TEST_FILE> \
--gpu --output_vectors
```

`<log_dir>`: directory containing the checkpoint and options file. `<TEST_FILE>`: the test file path.

11.1 Principal component analysis

Use the PCA class implemented in scikit-learn to do the principal component analysis (PCA) on the SPANet embedding vectors. The variance ratio of the first ten components is shown in Figure 7.

Calculate the correlation coefficients with principal components and the high-level observables. The high-level observables are the DNN input features that are constructed by the SPANet2 pairing.

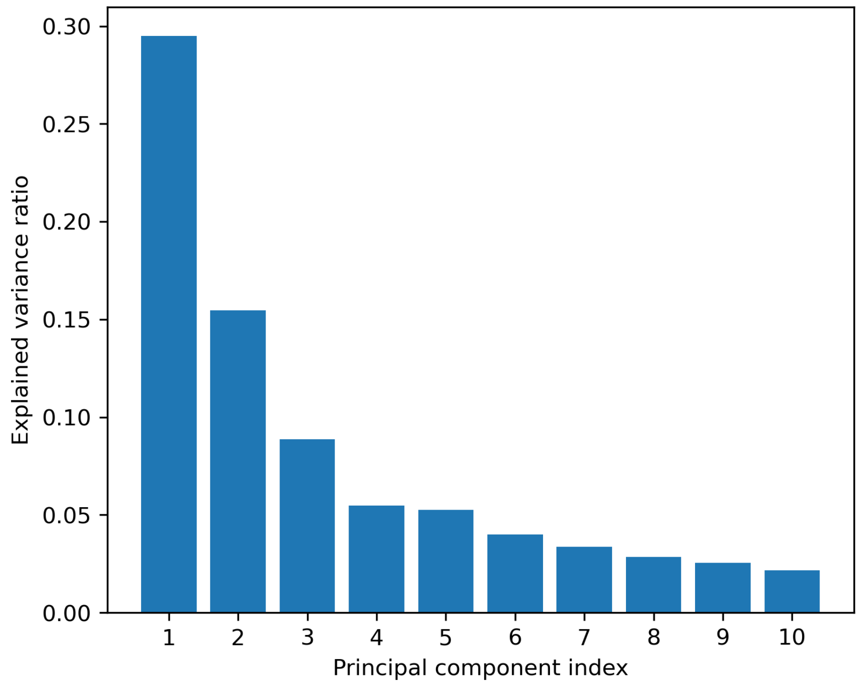


Figure 7: The variance ratio of the first ten principal components.

The results are presented in Figure 8. In Figure 9, the correlation coefficients of signal and background events are calculated separately. The level of correlation of most variables is very low in the background case compared to the signal one's.

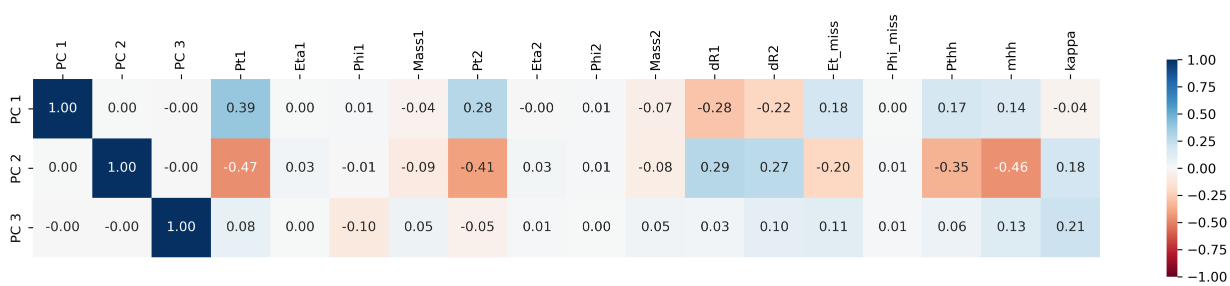


Figure 8: The correlation coefficients of the first three principal components and high-level observables.

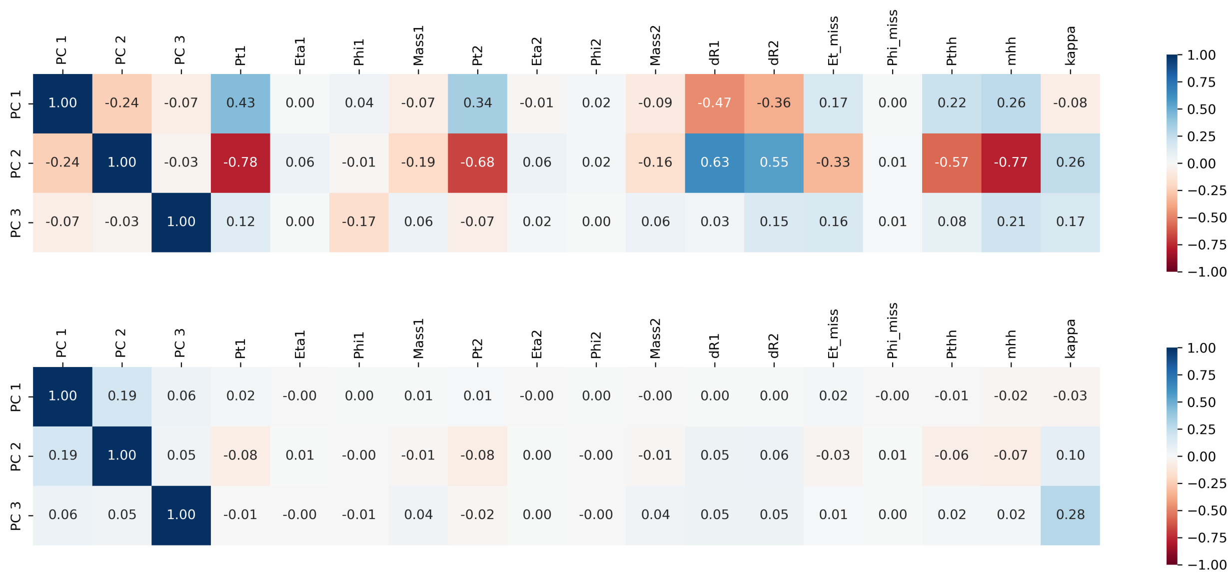


Figure 9: The correlation coefficients of the first three principal components and high-level observables. Where the signal and background samples are calculated separately.