

```
In [1]: %cd ../../
```

/home/jan/FMF/masters

Get generator pipeline + classification

```
In [2]: from ml_hep_sim.analysis.generator_pipeline import get_generator_pipeline
```

```
In [3]: class_pipeline = get_generator_pipeline(use_classifier=True)
```

Use cut on classifier

```
In [4]: from ml_hep_sim.pipeline.blocks import CutBlock, CutByIndexBlock  
from ml_hep_sim.pipeline.pipes import Pipeline
```

```
In [5]: b_sig_gen_class = class_pipeline.pipes[-4]  
b_bkg_gen_class = class_pipeline.pipes[-3]
```

```
b_sig_mc_class = class_pipeline.pipes[-2]  
b_bkg_mc_class = class_pipeline.pipes[-1]
```

```
In [6]: cut_value = 0.5
```

```
b_sig_gen_class_cut = CutBlock(cut_value)(b_sig_gen_class)  
b_bkg_gen_class_cut = CutBlock(cut_value)(b_bkg_gen_class)
```

```
b_sig_mc_class_cut = CutBlock(cut_value)(b_sig_mc_class)  
b_bkg_mc_class_cut = CutBlock(cut_value)(b_bkg_mc_class)
```

Cut all events

```
In [7]: from ml_hep_sim.pipeline.blocks import CutByIndexBlock
```

```
In [8]: b_sig_gen_data = class_pipeline.pipes[-8]  
b_bkg_gen_data = class_pipeline.pipes[-5]
```

```
b_sig_mc_data = class_pipeline.pipes[1]  
b_bkg_mc_data = class_pipeline.pipes[3]
```

```
In [9]: b_sig_gen_data_cut = CutByIndexBlock()(b_sig_gen_class_cut, b_sig_gen_data)  
b_bkg_gen_data_cut = CutByIndexBlock()(b_bkg_gen_class_cut, b_bkg_gen_data)
```

```
b_sig_mc_data_cut = CutByIndexBlock()(b_sig_mc_class_cut, b_sig_mc_data)  
b_bkg_mc_data_cut = CutByIndexBlock()(b_bkg_mc_class_cut, b_bkg_mc_data)
```

Rescale back to original

```
In [10]: from ml_hep_sim.pipeline.blocks import RedoRescaleDataBlock
```

```
In [11]: b_sig_gen_data_cut_rescale = RedoRescaleDataBlock(scaler_idx=0)(class_pipeline.pipes[7], b_sig_gen_data_cut)  
b_bkg_gen_data_cut_rescale = RedoRescaleDataBlock(scaler_idx=0)(class_pipeline.pipes[10], b_bkg_gen_data_cut)
```

```
b_sig_mc_data_cut_rescale = RedoRescaleDataBlock(scaler_idx=-1)(class_pipeline.pipes[1], b_sig_mc_data_cut)  
b_bkg_mc_data_cut_rescale = RedoRescaleDataBlock(scaler_idx=-1)(class_pipeline.pipes[3], b_bkg_mc_data_cut)
```

Do fit

```
In [12]: pipe = Pipeline()  
pipe.compose(  
    class_pipeline,  
    b_sig_gen_class_cut,  
    b_bkg_gen_class_cut,  
    b_sig_mc_class_cut,  
    b_bkg_mc_class_cut,  
    b_sig_gen_data_cut,  
    b_bkg_gen_data_cut,  
    b_sig_mc_data_cut,  
    b_bkg_mc_data_cut,  
    b_sig_gen_data_cut_rescale,
```

```

    b_bkg_gen_data_cut_rescale,
    b_sig_mc_data_cut_rescale,
    b_bkg_mc_data_cut_rescale,
)
pipe.fit()

```

```

100%|██████████| 10/10 [00:05<00:00, 1.92it/s]
100%|██████████| 10/10 [00:05<00:00, 1.92it/s]
100%|██████████| 10/10 [00:00<00:00, 3565.68it/s]
100%|██████████| 10/10 [00:00<00:00, 2044.80it/s]
100%|██████████| 10/10 [00:00<00:00, 146.63it/s]
100%|██████████| 10/10 [00:00<00:00, 148.04it/s]

```

```
Out[12]: <ml_hep_sim.pipeline.pipes.Pipeline at 0x7fcbb2f72e50>
```

```
In [13]: pipe.draw_pipeline_tree(to_graphviz_file="pipeline_gen_cut", block_idx=-4)
```

```
Out[13]: <treelib.tree.Tree at 0x7fcbb2f721f0>
```

Plot classifier cut

```

In [14]: import matplotlib.pyplot as plt
from ml_hep_sim.plotting.style import style_setup, set_size
from ml_hep_sim.stats.stat_plots import N_sample_plot
from ml_hep_sim.data_utils.higgs.process_higgs_dataset import LATEX_COLNAMES

set_size()
style_setup(seaborn_pallete=True)

```

```

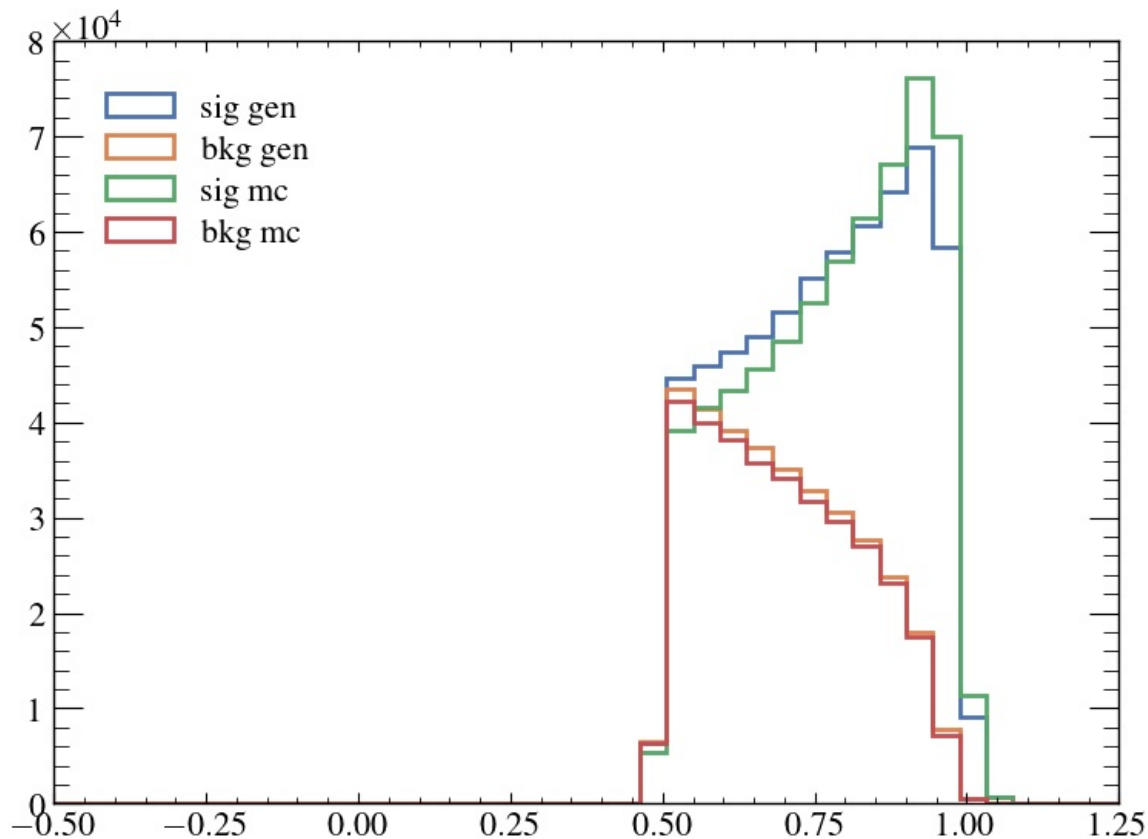
In [15]: sig_gen = pipe.pipes[-4-8].results
bkg_gen = pipe.pipes[-3-8].results
sig_mc = pipe.pipes[-2-8].results[: len(sig_gen)]
bkg_mc = pipe.pipes[-1-8].results[: len(sig_gen)]

```

```

In [16]: plt.hist(sig_gen, histtype="step", range=(-0.5, 1.25), bins=40, lw=2)
plt.hist(bkg_gen, histtype="step", range=(-0.5, 1.25), bins=40, lw=2)
plt.hist(sig_mc, histtype="step", range=(-0.5, 1.25), bins=40, lw=2)
plt.hist(bkg_mc, histtype="step", range=(-0.5, 1.25), bins=40, lw=2)
plt.legend(["sig gen", "bkg gen", "sig mc", "bkg mc"], loc="upper left")
plt.tight_layout()

```



```

In [17]: sig_mc = pipe.pipes[-2].reference_data
bkg_mc = pipe.pipes[-1].reference_data
sig_gen = pipe.pipes[-4].generated_data
bkg_gen = pipe.pipes[-3].generated_data

```

```
In [18]: BIN_RANGES = [
```

```

[0, 4],
[-3, 3],
[-0.1, 4],
[0, 5],
[-4, 4],
[0, 4],
[-5, 5],
[0, 5],
[-4, 4],
[0, 5],
[-3, 3],
[0, 3],
[0, 3],
[0.9, 1.5],
[0, 3],
[0, 2.5],
[0, 2.5],
[0, 2],
]

```

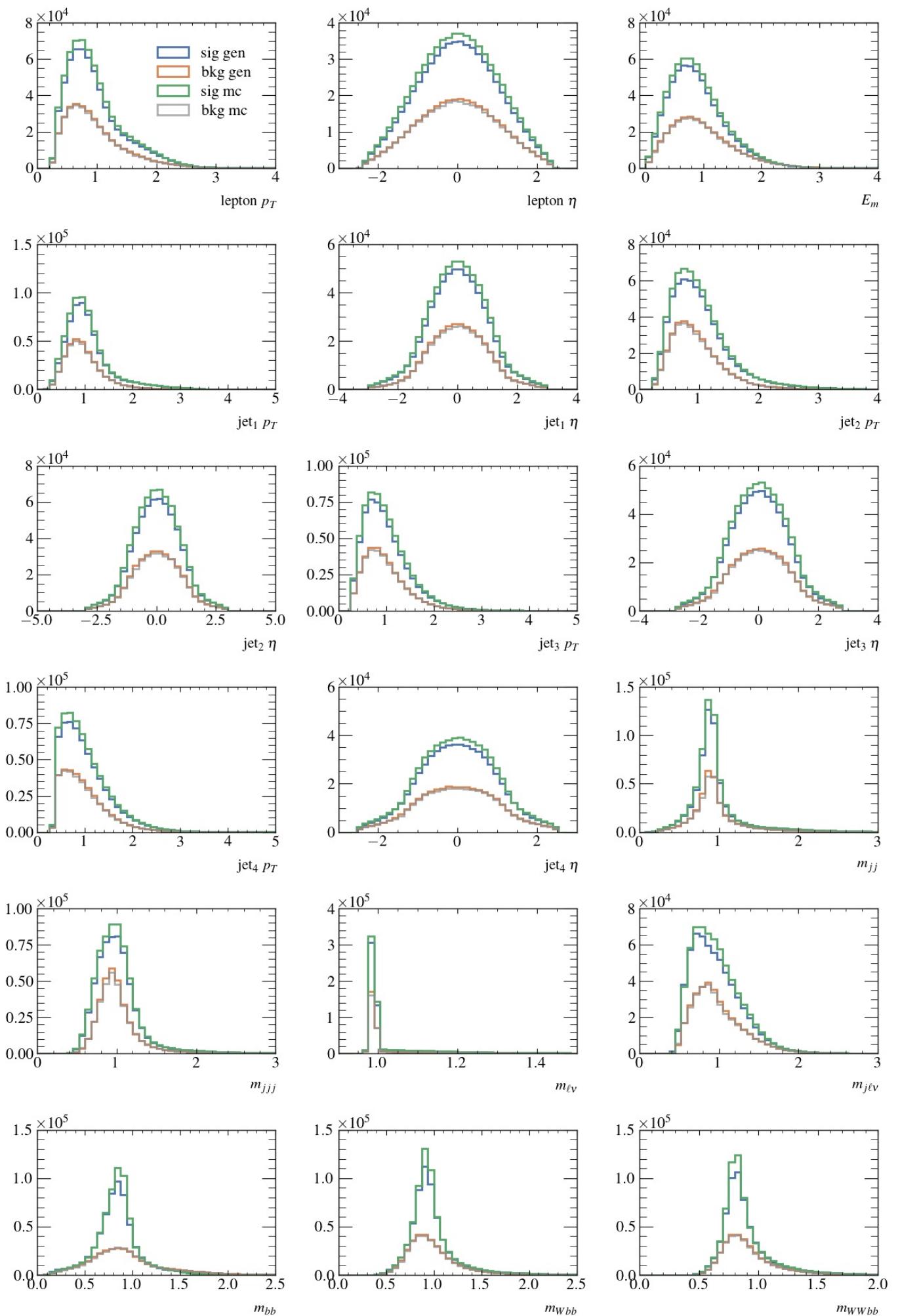
```

In [19]: fig, axs = plt.subplots(6, 3, figsize=(13, 19))
axs = axs.flatten()

res = [sig_gen, bkg_gen, sig_mc, bkg_mc]

N_sample_plot(res, axs, n_bins=40, log_scale=False,
              labels=LATEX_COLNAMES, lw=2, alpha=1,
              label=["sig gen", "bkg gen", "sig mc", "bkg mc"],
              xlim=BIN_RANGES, bin_range=BIN_RANGES)
plt.tight_layout()
plt.savefig("gen_mc_cut_dists.pdf")

```



In []:

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js