

```
In [1]: %cd ../../
/home/jan/FMF/masters

In [2]: saved = "ml_hep_sim/analysis/results/cls/"

$q_0$ is used to test  $\mu=0$  in a class of model where we assume  $\mu \geq 0$ . Rejecting  $\mu=0$  hypothesis leads to the discovery of a new signal.
```

Get CLs pipeline and fit

```
In [3]: from ml_hep_sim.analysis.cls_pipeline import get_cls_pipeline

import matplotlib.pyplot as plt
from ml_hep_sim.plotting.style import style_setup, set_size

import numpy as np

from ml_hep_sim.pipeline.pipeline_loggers import setup_logger

logger = setup_logger(log_name="cls", log_path="ml_pipeline/")

set_size()
style_setup(seaborn_pallette=True)

In [4]: use_class = False
pts = 40

cls_pipeline = get_cls_pipeline(pts=pts, lumi=100, use_classifier=use_class,
                                 bin_range=(0.5, 1.1) if use_class else (0.01, 3.0),
                                 N_gen=10**6, logger=logger, scale_by_alpha=False)

WARNING:root:available variables: {'lepton pT': 0, 'lepton eta': 1, 'missing energy': 2, 'jet1 pT': 3, 'jet1 eta': 4, 'jet2 pT': 5, 'jet2 eta': 6, 'jet3 pT': 7, 'jet3 eta': 8, 'jet4 pT': 9, 'jet4 eta': 10, 'm jj': 11, 'm jjj': 12, 'm lv': 13, 'm jlv': 14, 'm bb': 15, 'm wbb': 16, 'm wwbb': 17}

In [5]: if use_class:
    saved += "class_"
else:
    saved += "mbb_"

In [6]: saved
Out[6]: 'ml_hep_sim/analysis/results/cls/mbb_'

In [7]: res = cls_pipeline.fit()

INFO:ml_hep_sim.pipeline.pipeline_loggers:fitting #0: <ml_hep_sim.pipeline.blocks.DatasetBuilderBlock object at 0x7f4b886b86a0>!
INFO:ml_hep_sim.pipeline.pipeline_loggers:fitting #1: <ml_hep_sim.pipeline.blocks.ReferenceDataLoaderBlock object at 0x7f4b886b8490>!
INFO:ml_hep_sim.pipeline.pipeline_loggers:fitting #2: <ml_hep_sim.pipeline.blocks.DatasetBuilderBlock object at 0x7f4b70a05520>!
INFO:ml_hep_sim.pipeline.pipeline_loggers:fitting #3: <ml_hep_sim.pipeline.blocks.ReferenceDataLoaderBlock object at 0x7f4b70a054f0>!
INFO:ml_hep_sim.pipeline.pipeline_loggers:fitting #4: <ml_hep_sim.pipeline.blocks.ModelLoaderBlock object at 0x7f4b70a05550>!
INFO:ml_hep_sim.pipeline.pipeline_loggers:fitting #5: <ml_hep_sim.pipeline.blocks.ModelLoaderBlock object at 0x7f4b887151c0>!
INFO:ml_hep_sim.pipeline.pipeline_loggers:fitting #6: <ml_hep_sim.pipeline.blocks.DataGeneratorBlock object at 0x7f4b886b8730>!
INFO:ml_hep_sim.pipeline.pipeline_loggers:Generating 1000000 examples in 10 chunks of 100000 examples each using MADEMOGFlowModel.
100% |██████████| 10/10 [00:05<00:00, 1.89it/s]
INFO:ml_hep_sim.pipeline.pipeline_loggers:fitting #7: <ml_hep_sim.pipeline.blocks.GeneratedDataVerifierBlock object at 0x7f4b886b85e0>!
INFO:ml_hep_sim.pipeline.pipeline_loggers:Generated data check...
INFO:ml_hep_sim.pipeline.pipeline_loggers:nan OK
INFO:ml_hep_sim.pipeline.pipeline_loggers:pos-inf OK
INFO:ml_hep_sim.pipeline.pipeline_loggers:neg-inf OK
INFO:ml_hep_sim.pipeline.pipeline_loggers:pos-inf or neg-inf OK
INFO:ml_hep_sim.pipeline.pipeline_loggers:pos-inf or neg-inf or nan OK
INFO:ml_hep_sim.pipeline.pipeline_loggers:fitting #8: <ml_hep_sim.pipeline.blocks.ModelLoaderBlock object at 0x7f4b886b8670>!
INFO:ml_hep_sim.pipeline.pipeline_loggers:fitting #9: <ml_hep_sim.pipeline.blocks.DataGeneratorBlock object at 0x7f4b886b86d0>!
```

```
x7f4b886b8790>!
INFO:ml_hep_sim.pipeline.pipeline_loggers:Generating 1000000 examples in 10 chunks of 100000 examples each using MADEMOGFlowModel.
100%|██████████| 10/10 [00:05<00:00, 1.90it/s]
INFO:ml_hep_sim.pipeline.pipeline_loggers:fitting #10: <ml_hep_sim.pipeline.blocks.GeneratedDataVerifierBlock object at 0x7f4b886b88b0>!
INFO:ml_hep_sim.pipeline.pipeline_loggers:Generated data check...
INFO:ml_hep_sim.pipeline.pipeline_loggers:nan OK
INFO:ml_hep_sim.pipeline.pipeline_loggers:pos-inf OK
INFO:ml_hep_sim.pipeline.pipeline_loggers:neg-inf OK
INFO:ml_hep_sim.pipeline.pipeline_loggers:pos-inf or neg-inf OK
INFO:ml_hep_sim.pipeline.pipeline_loggers:pos-inf or neg-inf or nan OK
INFO:ml_hep_sim.pipeline.pipeline_loggers:fitting #11: <ml_hep_sim.pipeline.blocks.ClassifierRunnerBlock object at 0x7f4b70a05970>!
100%|██████████| 10/10 [00:00<00:00, 6048.90it/s]
INFO:ml_hep_sim.pipeline.pipeline_loggers:fitting #12: <ml_hep_sim.pipeline.blocks.ClassifierRunnerBlock object at 0x7f4b70a05940>!
100%|██████████| 10/10 [00:00<00:00, 5726.01it/s]
INFO:ml_hep_sim.pipeline.pipeline_loggers:fitting #13: <ml_hep_sim.pipeline.blocks.ClassifierRunnerBlock object at 0x7f4b70a059a0>!
100%|██████████| 10/10 [00:00<00:00, 363.46it/s]
INFO:ml_hep_sim.pipeline.pipeline_loggers:fitting #14: <ml_hep_sim.pipeline.blocks.ClassifierRunnerBlock object at 0x7f4b70a059d0>!
100%|██████████| 10/10 [00:00<00:00, 374.83it/s]
INFO:ml_hep_sim.pipeline.pipeline_loggers:fitting #15: <ml_hep_sim.pipeline.blocks.CutBlock object at 0x7f4adaf62f70>!
INFO:ml_hep_sim.pipeline.pipeline_loggers:fitting #16: <ml_hep_sim.pipeline.blocks.CutBlock object at 0x7f4b88168fa0>!
INFO:ml_hep_sim.pipeline.pipeline_loggers:fitting #17: <ml_hep_sim.pipeline.blocks.CutBlock object at 0x7f4b8816f2b0>!
INFO:ml_hep_sim.pipeline.pipeline_loggers:fitting #18: <ml_hep_sim.pipeline.blocks.CutBlock object at 0x7f4b88715e80>!
INFO:ml_hep_sim.pipeline.pipeline_loggers:fitting #19: <ml_hep_sim.pipeline.blocks.CutByIndexBlock object at 0x7f4b88715e20>!
INFO:ml_hep_sim.pipeline.pipeline_loggers:fitting #20: <ml_hep_sim.pipeline.blocks.CutByIndexBlock object at 0x7f4b88715dc0>!
INFO:ml_hep_sim.pipeline.pipeline_loggers:fitting #21: <ml_hep_sim.pipeline.blocks.CutByIndexBlock object at 0x7f4b88715ac0>!
INFO:ml_hep_sim.pipeline.pipeline_loggers:fitting #22: <ml_hep_sim.pipeline.blocks.CutByIndexBlock object at 0x7f4b88715bb0>!
INFO:ml_hep_sim.pipeline.pipeline_loggers:fitting #23: <ml_hep_sim.pipeline.blocks.RedoRescaleDataBlock object at 0x7f4b88715f40>!
INFO:ml_hep_sim.pipeline.pipeline_loggers:fitting #24: <ml_hep_sim.pipeline.blocks.RedoRescaleDataBlock object at 0x7f4b886b8580>!
INFO:ml_hep_sim.pipeline.pipeline_loggers:fitting #25: <ml_hep_sim.pipeline.blocks.RedoRescaleDataBlock object at 0x7f4b886b86d0>!
INFO:ml_hep_sim.pipeline.pipeline_loggers:fitting #26: <ml_hep_sim.pipeline.blocks.RedoRescaleDataBlock object at 0x7f4b886b84c0>!
INFO:ml_hep_sim.pipeline.pipeline_loggers:fitting #27: <ml_hep_sim.pipeline.blocks.VariableExtractBlock object at 0x7f4b70a056a0>!
INFO:ml_hep_sim.pipeline.pipeline_loggers:fitting #28: <ml_hep_sim.pipeline.blocks.VariableExtractBlock object at 0x7f4b70a05760>!
INFO:ml_hep_sim.pipeline.pipeline_loggers:fitting #29: <ml_hep_sim.pipeline.blocks.VariableExtractBlock object at 0x7f4b70a05b20>!
INFO:ml_hep_sim.pipeline.pipeline_loggers:fitting #30: <ml_hep_sim.pipeline.blocks.VariableExtractBlock object at 0x7f4b70a05be0>!
INFO:ml_hep_sim.pipeline.pipeline_loggers:fitting #31: <ml_hep_sim.analysis.utils.SigBkgBlock object at 0x7f4b70a05af0>!
INFO:ml_hep_sim.pipeline.pipeline_loggers:fitting #32: <ml_hep_sim.analysis.hists_pipeline.MakeHistsFromSamplesLumi object at 0x7f4b70a05a90>!
DEBUG:ml_hep_sim.pipeline.pipeline_loggers:considering: S_mc=10.0, B_mc=1000 and alpha=0.01 with 100.0 lumi
INFO:ml_hep_sim.pipeline.pipeline_loggers:fitting #33: <ml_hep_sim.analysis.cls_pipeline.ClsBlock object at 0x7f4b70a05bb0>!
INFO:ml_hep_sim.pipeline.pipeline_loggers:fitting #34: <ml_hep_sim.analysis.hists_pipeline.MakeHistsFromSamplesLumi object at 0x7f4b70a05e20>!
DEBUG:ml_hep_sim.pipeline.pipeline_loggers:considering: S_mc=10.0, B_mc=1000 and alpha=0.01 with 100.0 lumi
INFO:ml_hep_sim.pipeline.pipeline_loggers:fitting #35: <ml_hep_sim.analysis.cls_pipeline.ClsBlock object at 0x7f4b70a05df0>!
INFO:ml_hep_sim.pipeline.pipeline_loggers:fitting #36: <ml_hep_sim.analysis.hists_pipeline.MakeHistsFromSamplesLumi object at 0x7f4b70a05d90>!
DEBUG:ml_hep_sim.pipeline.pipeline_loggers:considering: S_mc=10.0, B_mc=1000 and alpha=0.01 with 100.0 lumi
INFO:ml_hep_sim.pipeline.pipeline_loggers:fitting #37: <ml_hep_sim.analysis.cls_pipeline.ClsBlock object at 0x7f4b70a05dc0>!
```



```
sLumi object at 0x7f4b5b9434f0>
DEBUG:ml_hep_sim.pipeline.pipeline_loggers:considering: S_mc=100.0, B_mc=1000 and alpha=0.1 with 100.0 lumi
INFO:ml_hep_sim.pipeline.pipeline_loggers:fitting #6423: <ml_hep_sim.analysis.cls_pipeline.CLSBlock object at 0x7f4b5b943550>!
INFO:ml_hep_sim.pipeline.pipeline_loggers:fitting #6424: <ml_hep_sim.analysis.hists_pipeline.MakeHistsFromSample sLumi object at 0x7f4b5b9435b0>!
DEBUG:ml_hep_sim.pipeline.pipeline_loggers:considering: S_mc=100.0, B_mc=1000 and alpha=0.1 with 100.0 lumi
INFO:ml_hep_sim.pipeline.pipeline_loggers:fitting #6425: <ml_hep_sim.analysis.cls_pipeline.CLSBlock object at 0x7f4b5b943610>!
INFO:ml_hep_sim.pipeline.pipeline_loggers:fitting #6426: <ml_hep_sim.analysis.hists_pipeline.MakeHistsFromSample sLumi object at 0x7f4b5b943670>!
DEBUG:ml_hep_sim.pipeline.pipeline_loggers:considering: S_mc=100.0, B_mc=1000 and alpha=0.1 with 100.0 lumi
INFO:ml_hep_sim.pipeline.pipeline_loggers:fitting #6427: <ml_hep_sim.analysis.cls_pipeline.CLSBlock object at 0x7f4b5b9436d0>!
INFO:ml_hep_sim.pipeline.pipeline_loggers:fitting #6428: <ml_hep_sim.analysis.hists_pipeline.MakeHistsFromSample sLumi object at 0x7f4b5b943730>!
DEBUG:ml_hep_sim.pipeline.pipeline_loggers:considering: S_mc=100.0, B_mc=1000 and alpha=0.1 with 100.0 lumi
INFO:ml_hep_sim.pipeline.pipeline_loggers:fitting #6429: <ml_hep_sim.analysis.cls_pipeline.CLSBlock object at 0x7f4b5b943790>!
INFO:ml_hep_sim.pipeline.pipeline_loggers:fitting #6430: <ml_hep_sim.analysis.hists_pipeline.MakeHistsFromSample sLumi object at 0x7f4b5b9437f0>!
DEBUG:ml_hep_sim.pipeline.pipeline_loggers:considering: S_mc=100.0, B_mc=1000 and alpha=0.1 with 100.0 lumi
INFO:ml_hep_sim.pipeline.pipeline_loggers:fitting #6431: <ml_hep_sim.analysis.cls_pipeline.CLSBlock object at 0x7f4b5b943850>!
INFO:ml_hep_sim.pipeline.pipeline_loggers:fitting #6432: <ml_hep_sim.analysis.cls_pipeline.CLSBlockResultsParser object at 0x7f4b886b84f0>!
```

```
In [8]: parsed_res = res.pipes[-1].parsed_results
mc_res = parsed_res["mc_res"]
ml_res = parsed_res["ml_res"]
```

```
In [9]: ml_res
```

	sig_frac	bkg_err	p_sb	p_b	p_s	p_exp_sb	p_exp_b	
0	0.01	0.010000	0.42971033308692386	0.571927597749965	0.7513369433079611	0.3600236782686039	0.5	0.7
1	0.01	0.012308	0.4327485985332032	0.5740003675861907	0.7539169362434643	0.36093944417075197	0.5	0.7
2	0.01	0.014615	0.43217428508189026	0.57339290457229	0.7537140443066022	0.3609726295034114	0.5	0.7
3	0.01	0.016923	0.4326701247595891	0.5742557571235596	0.7534449927447462	0.3606208345727053	0.5	0.7
4	0.01	0.019231	0.43728470670208336	0.5766906040821187	0.7582657036663208	0.36268484752692476	0.5	0.7
...
1595	0.10	0.090769	0.01834011758576407	0.6230791729518139	0.029434650333244904	0.008133187630173715	0.5	0.01
1596	0.10	0.093077	0.018178946270706216	0.620901050701423	0.029278330661817568	0.008180749504565198	0.5	0.016
1597	0.10	0.095385	0.02186976396491997	0.6292728524299578	0.03475402423681422	0.009475167728140945	0.5	0.01
1598	0.10	0.097692	0.022880748981502014	0.6271929926028323	0.036481193590106265	0.010116074604603136	0.5	0.02
1599	0.10	0.100000	0.025379625579974512	0.6284897979935578	0.04038192133109957	0.011263408049635786	0.5	0.02

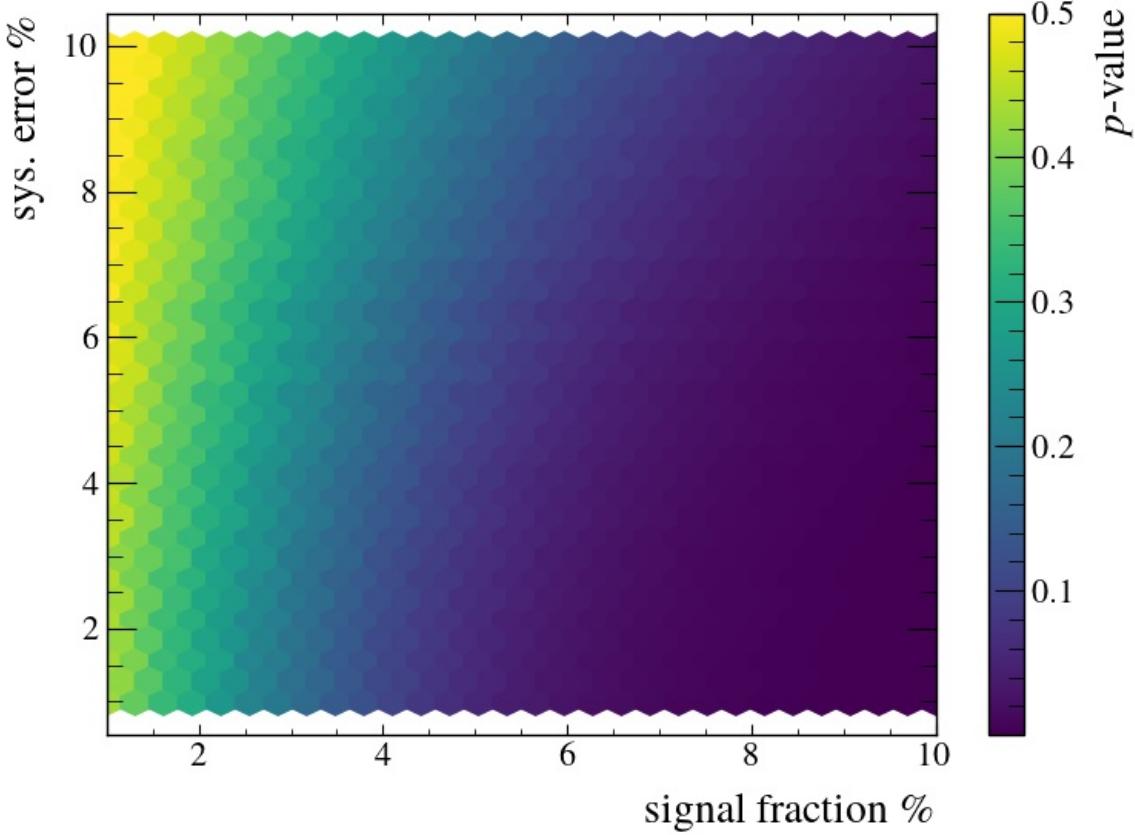
1600 rows × 9 columns

```
In [10]: x = np.array([float(i) * 100 for i in ml_res["sig_frac"].values])
y = np.array([float(i) * 100 for i in ml_res["bkg_err"].values])
z = np.array([float(i) for i in ml_res["p_sb"].values])

plt.xlabel("signal fraction %")
plt.ylabel("sys. error %")

plt.hexbin(x, y, z, gridsize=29)
plt.colorbar(label="$p\$-value")
plt.tight_layout()

plt.savefig(saved + "hexbin_clsb_ml.pdf")
```

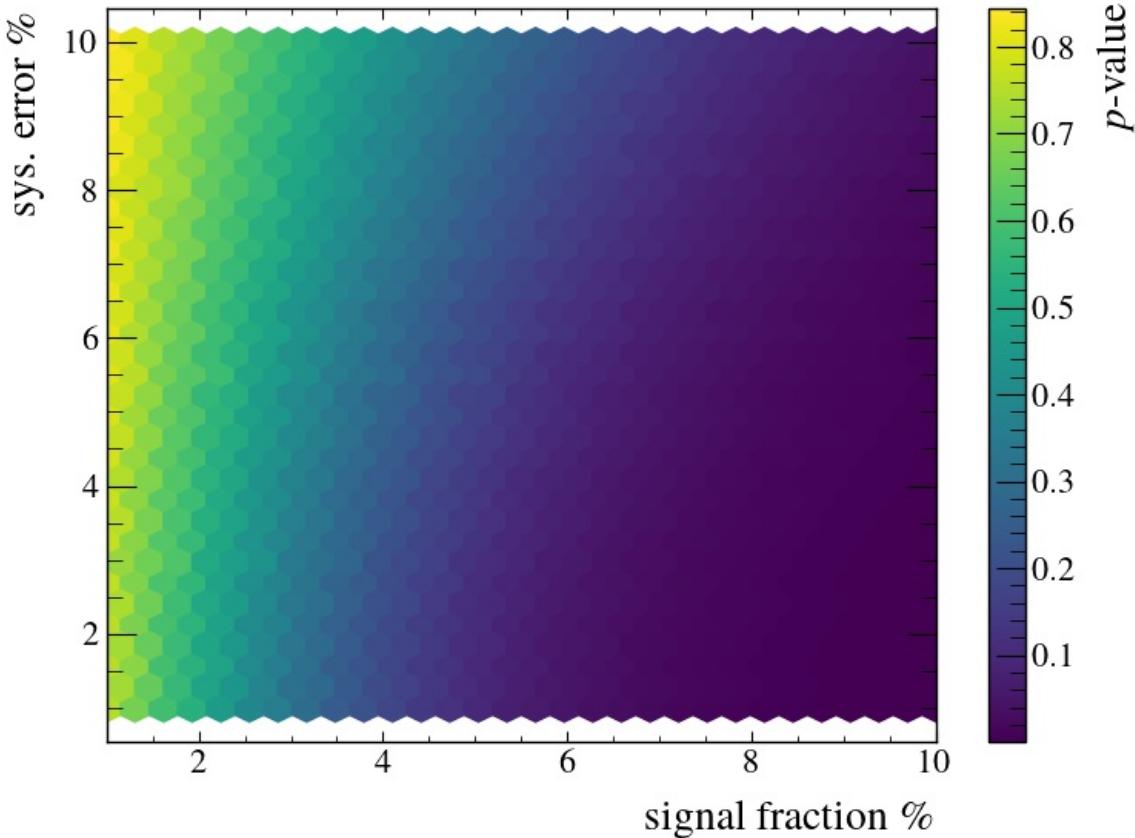


```
In [11]: x = np.array([float(i) * 100 for i in ml_res["sig_frac"].values])
y = np.array([float(i) * 100 for i in ml_res["bkg_err"].values])
z = np.array([float(i) for i in ml_res["p_s"].values])

plt.xlabel("signal fraction \\"%")
plt.ylabel("sys. error \\"%")

plt.hexbin(x, y, z, gridsize=29)
plt.colorbar(label="$p$-value")
plt.tight_layout()

plt.savefig(saved + "hexbin_cls_ml.pdf")
```



```
In [12]: x = np.array([float(i) * 100 for i in ml_res["sig_frac"].values])
y = np.array([float(i) * 100 for i in ml_res["bkg_err"].values])
z = np.array([float(i) for i in mc_res["p_sb"].values])
```

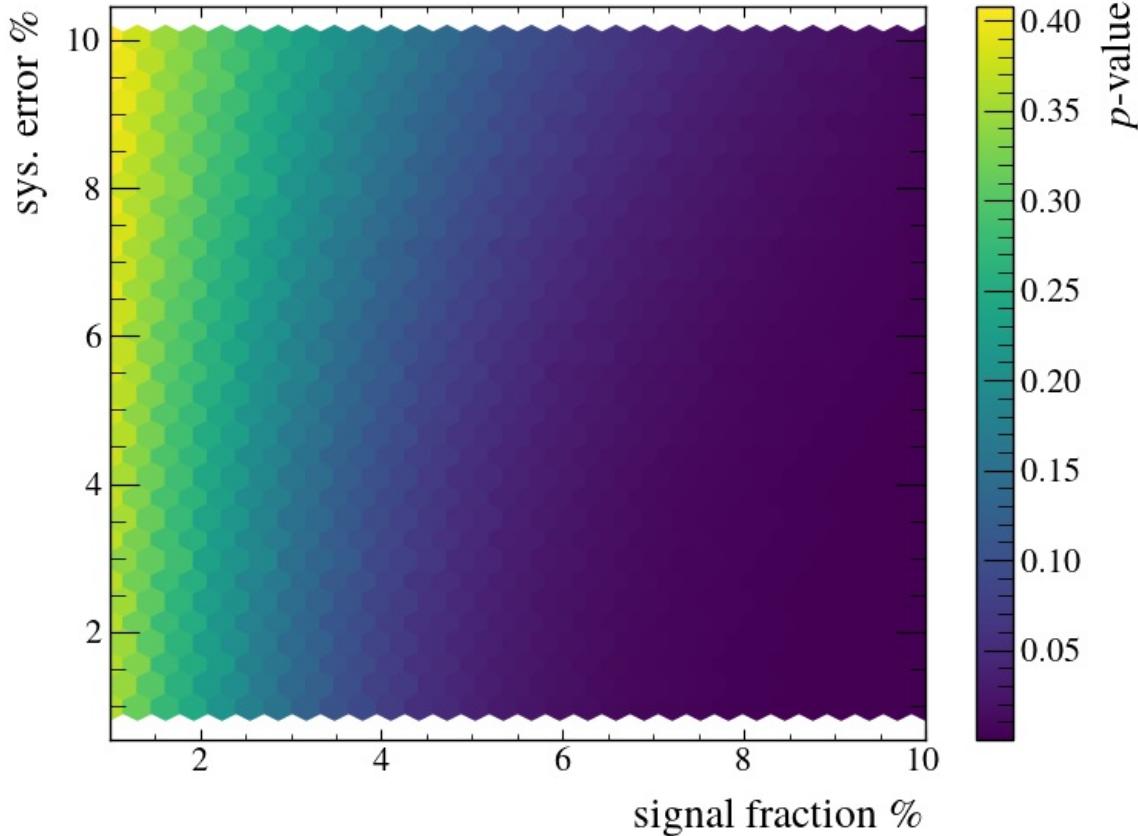
```

plt.xlabel("signal fraction \%")
plt.ylabel("sys. error \%")

plt.hexbin(x, y, z, gridsize=29)
plt.colorbar(label="$p\$-value")
plt.tight_layout()

plt.savefig(saved + "hexbin_clsb_mc.pdf")

```



```

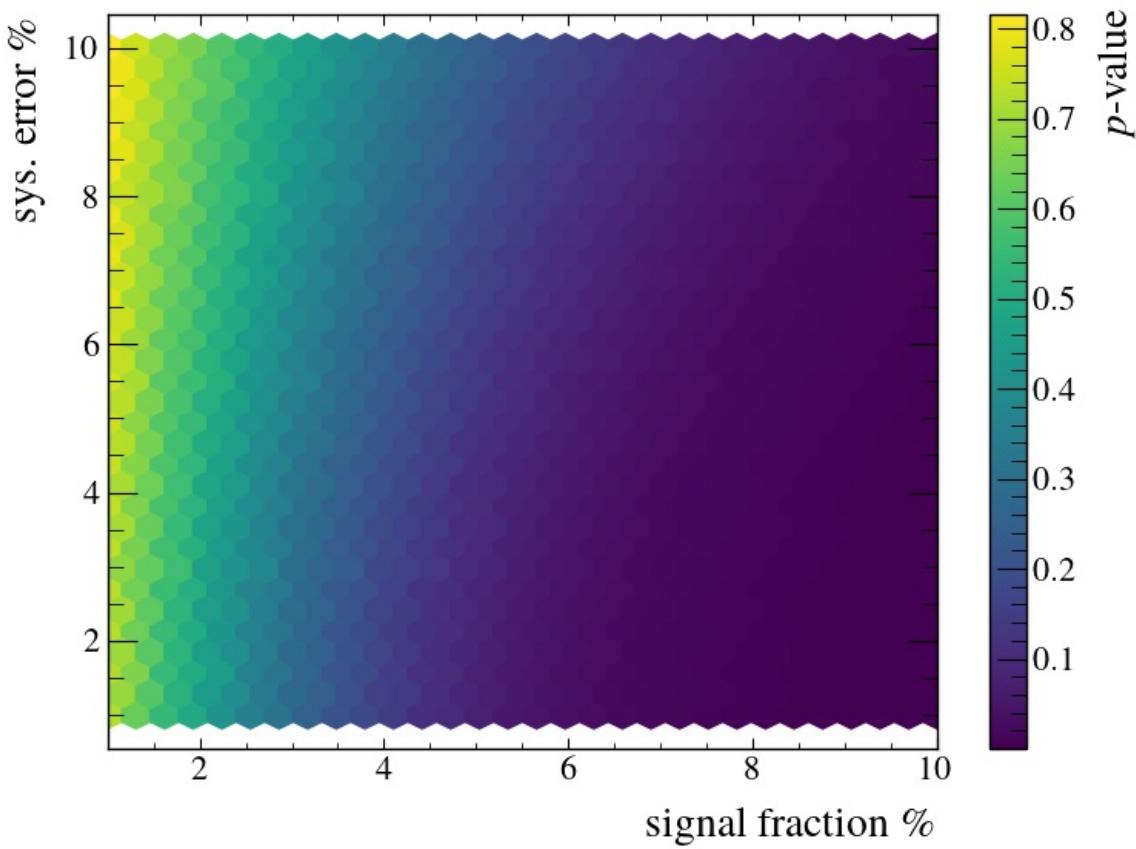
In [13]: x = np.array([float(i) * 100 for i in ml_res["sig_frac"].values])
y = np.array([float(i) * 100 for i in ml_res["bkg_err"].values])
z = np.array([float(i) for i in mc_res["p_s"].values])

plt.xlabel("signal fraction \%")
plt.ylabel("sys. error \%")

plt.hexbin(x, y, z, gridsize=29)
plt.colorbar(label="$p\$-value")
plt.tight_layout()

plt.savefig(saved + "hexbin_cls_mc.pdf")

```

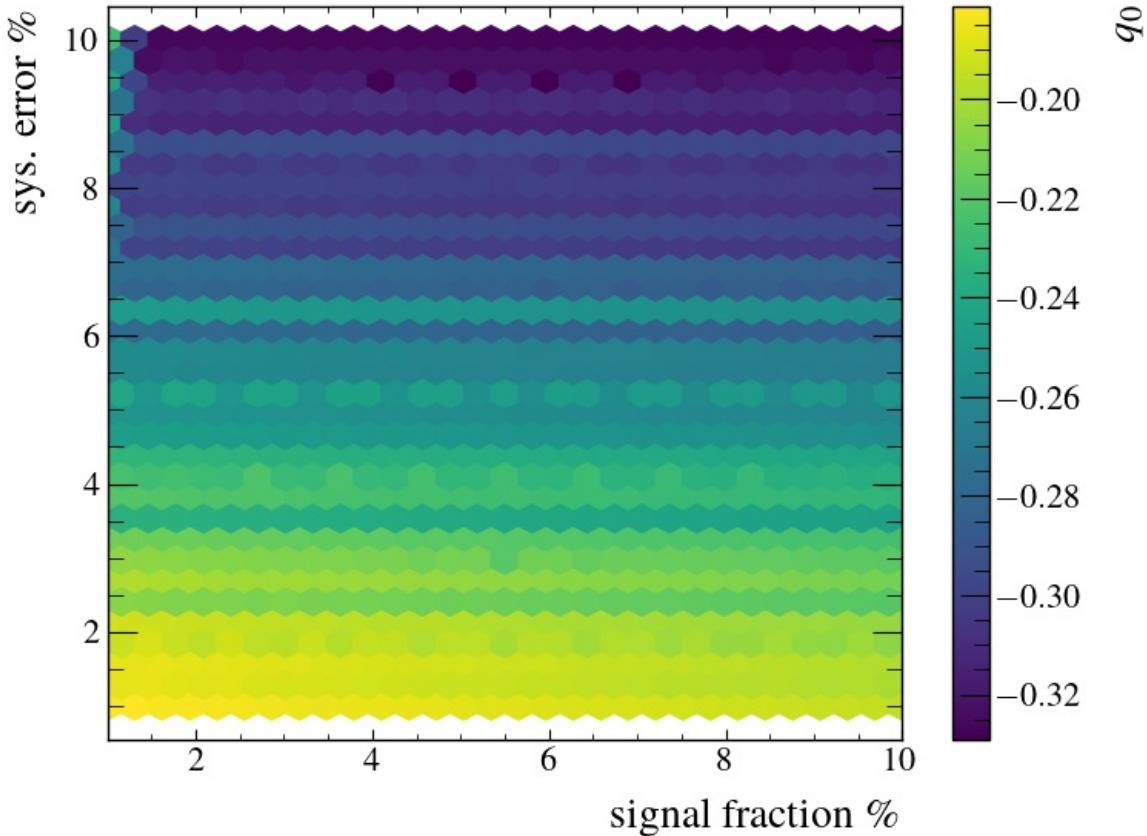


```
In [15]: x = np.array([float(i) * 100 for i in ml_res["sig_frac"].values])
y = np.array([float(i) * 100 for i in ml_res["bkg_err"].values])
z = np.array([float(i) for i in ml_res["teststat"].values])

plt.xlabel("signal fraction \\"%")
plt.ylabel("sys. error \\"%")

plt.hexbin(x, y, z, gridsize=29)
plt.colorbar(label="$q_0$")
plt.tight_layout()

plt.savefig(saved + "hexbin_q0_ml.pdf")
```



Constant signal fraction

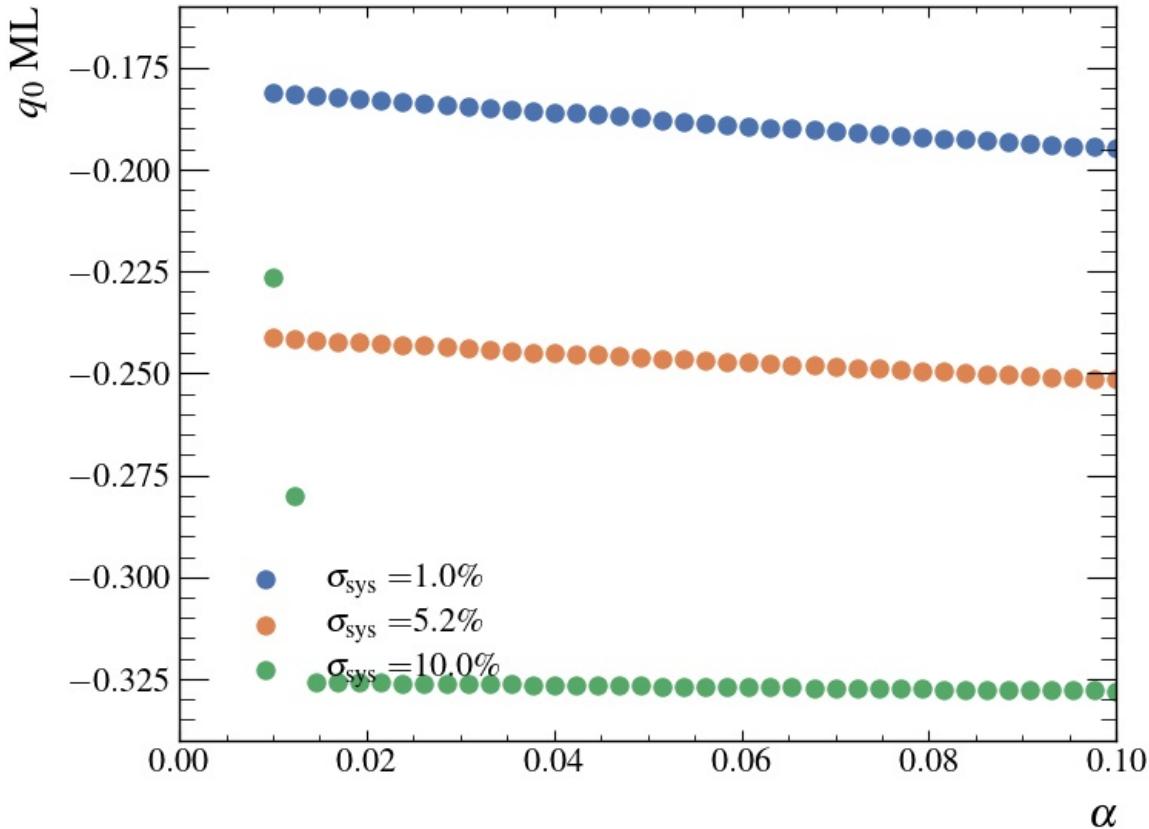
```
In [16]: sig_fracs = np.linspace(0.01, 0.1, pts)
bkg_errs = np.linspace(0.01, 0.1, pts)
```

```
In [17]: sig_fracs, bkg_errs
```

```
Out[17]: (array([0.01      , 0.01230769, 0.01461538, 0.01692308, 0.01923077,
       0.02153846, 0.02384615, 0.02615385, 0.02846154, 0.03076923,
       0.03307692, 0.03538462, 0.03769231, 0.04      , 0.04230769,
       0.04461538, 0.04692308, 0.04923077, 0.05153846, 0.05384615,
       0.05615385, 0.05846154, 0.06076923, 0.06307692, 0.06538462,
       0.06769231, 0.07      , 0.07230769, 0.07461538, 0.07692308,
       0.07923077, 0.08153846, 0.08384615, 0.08615385, 0.08846154,
       0.09076923, 0.09307692, 0.09538462, 0.09769231, 0.1      ]),
array([0.01      , 0.01230769, 0.01461538, 0.01692308, 0.01923077,
       0.02153846, 0.02384615, 0.02615385, 0.02846154, 0.03076923,
       0.03307692, 0.03538462, 0.03769231, 0.04      , 0.04230769,
       0.04461538, 0.04692308, 0.04923077, 0.05153846, 0.05384615,
       0.05615385, 0.05846154, 0.06076923, 0.06307692, 0.06538462,
       0.06769231, 0.07      , 0.07230769, 0.07461538, 0.07692308,
       0.07923077, 0.08153846, 0.08384615, 0.08615385, 0.08846154,
       0.09076923, 0.09307692, 0.09538462, 0.09769231, 0.1      ]))
```

```
In [18]: for e in [bkg_errs[0], bkg_errs[18], bkg_errs[-1]]:
    ml_ = ml_res[ml_res["bkg_err"] == e]
    plt.scatter(sig_fracs, ml_[ "teststat"], label="$\sigma_{\mathrm{sys}}=$" + f"{100*e:.1f}\%")
plt.xlabel(r"$\alpha$")
plt.ylabel("$q_0$ ML")
plt.legend()
plt.tight_layout()

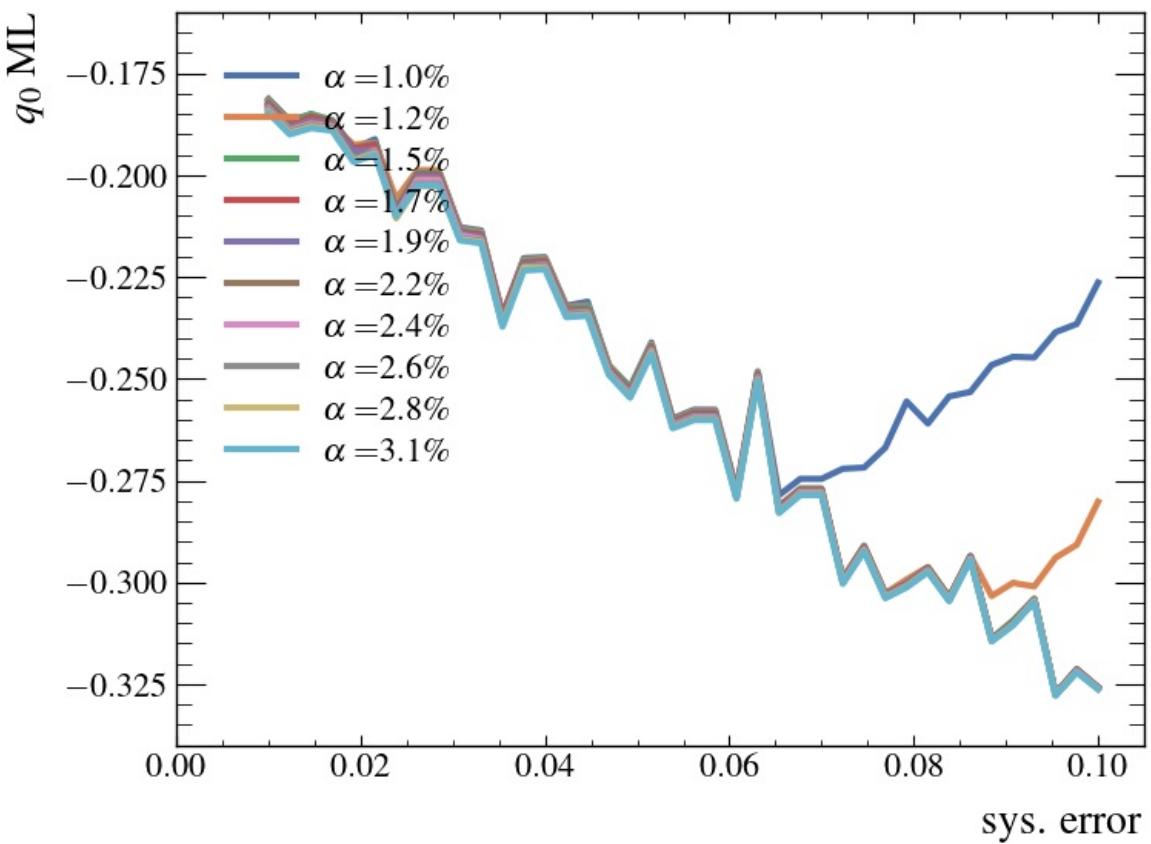
plt.savefig(saved + "q0_vs_sigfrac.pdf")
```



```
In [19]: for sf in [*sig_fracs[:10]]:
    ml_ = ml_res[ml_res["sig_frac"] == sf]
    plt.plot(bkg_errs, ml_[ "teststat"], label=r'$\alpha=' + f'{100*sf:.1f}\%', lw=3')

plt.legend(ncol=1, loc='upper left')
plt.xlim([0, 0.105])
plt.xlabel("sys. error")
plt.ylabel("$q_0$ ML")
plt.tight_layout()

plt.savefig(saved + "q0_vs_syserr.pdf")
```



```
In [20]: for i, f in enumerate(sig_fracs):
    ml_ = ml_res[ml_res["sig_frac"] == f]
    mc_ = mc_res[mc_res["sig_frac"] == f]

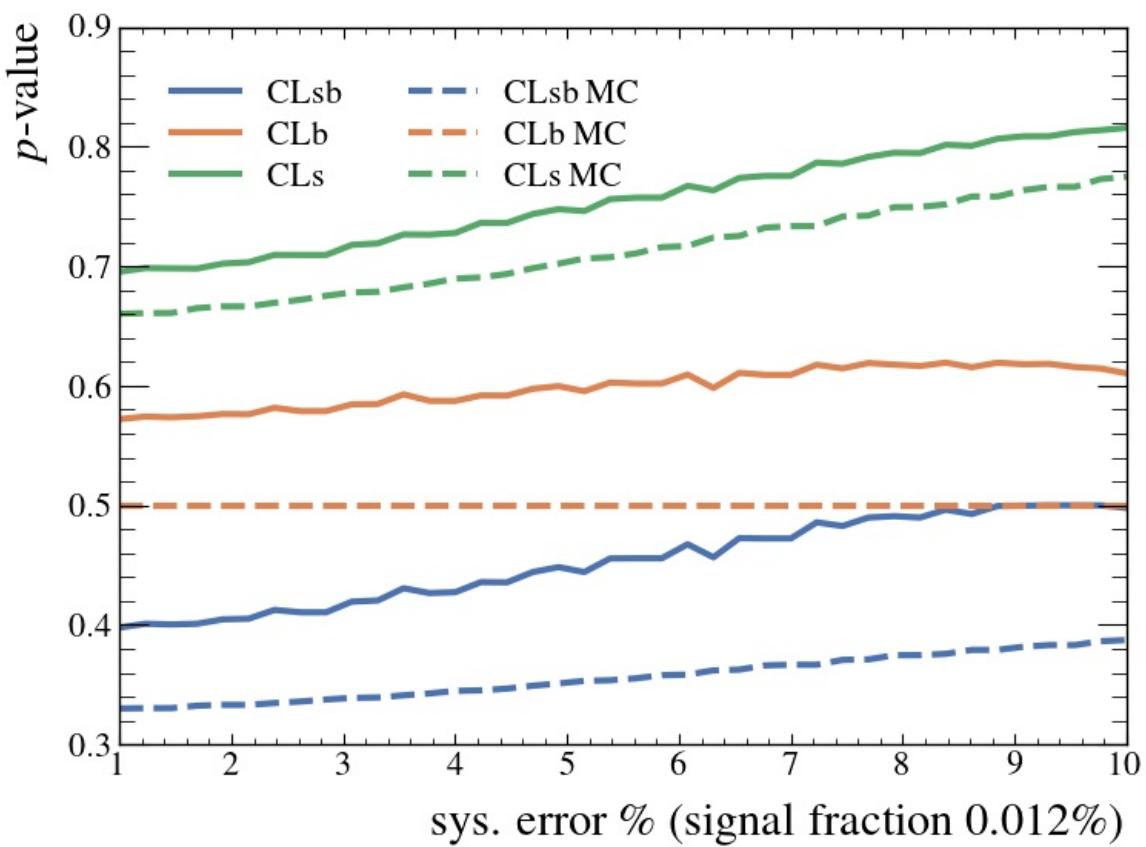
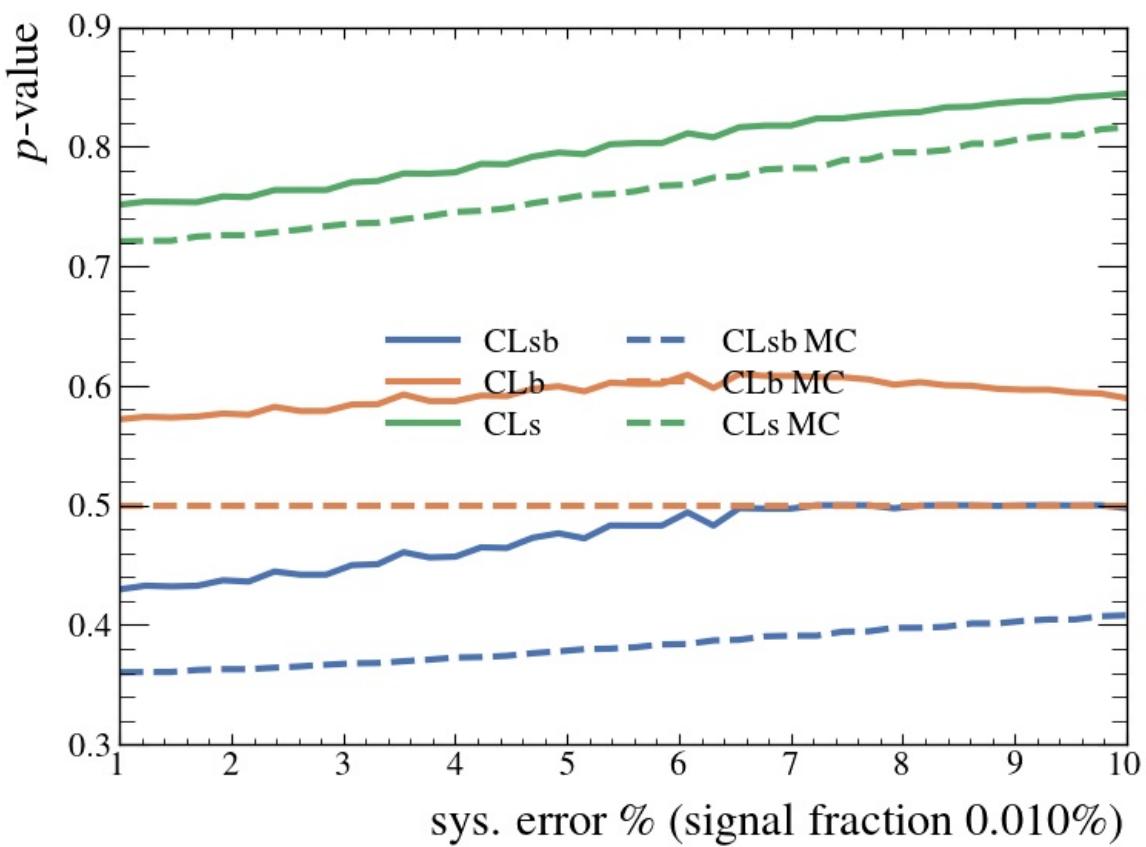
    plt.plot(ml_["bkg_err"] * 100, ml_["p_sb"], c="C0", lw=3)
    plt.plot(ml_["bkg_err"] * 100, ml_["p_b"], c="C1", lw=3)
    plt.plot(ml_["bkg_err"] * 100, ml_["p_s"], c="C2", lw=3)

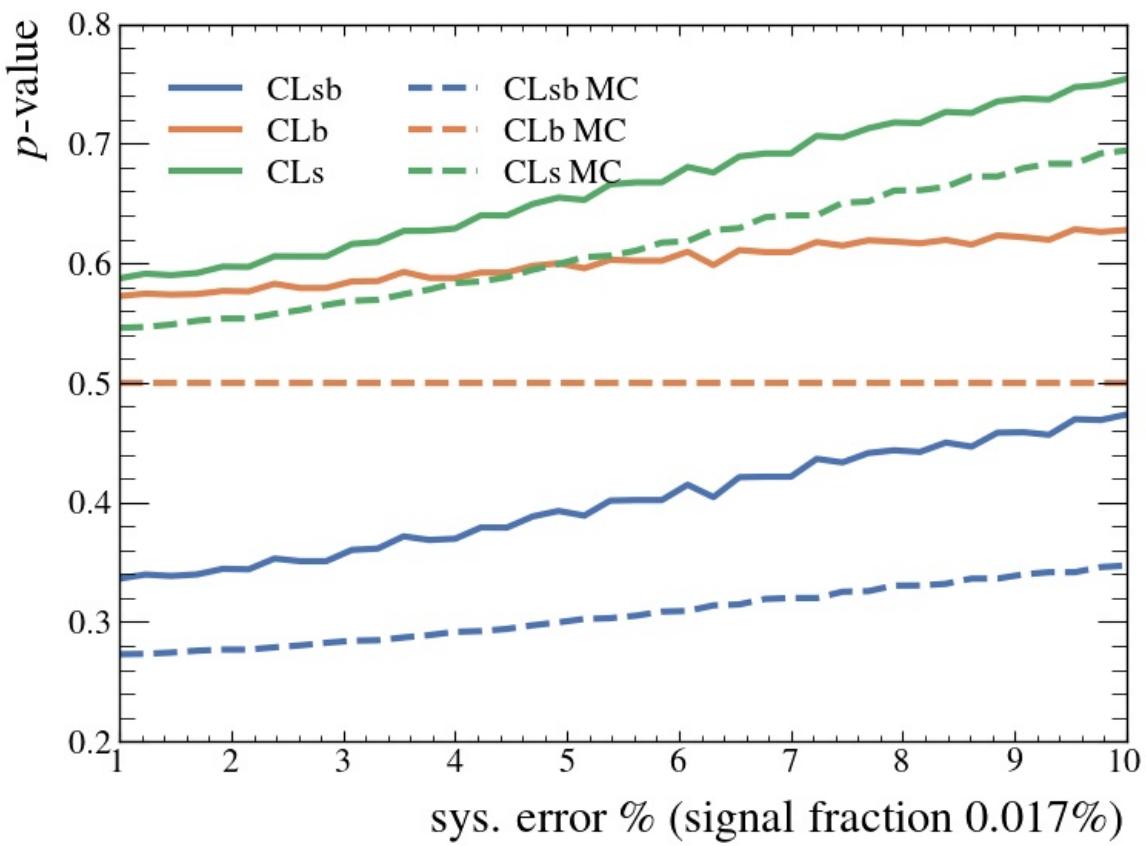
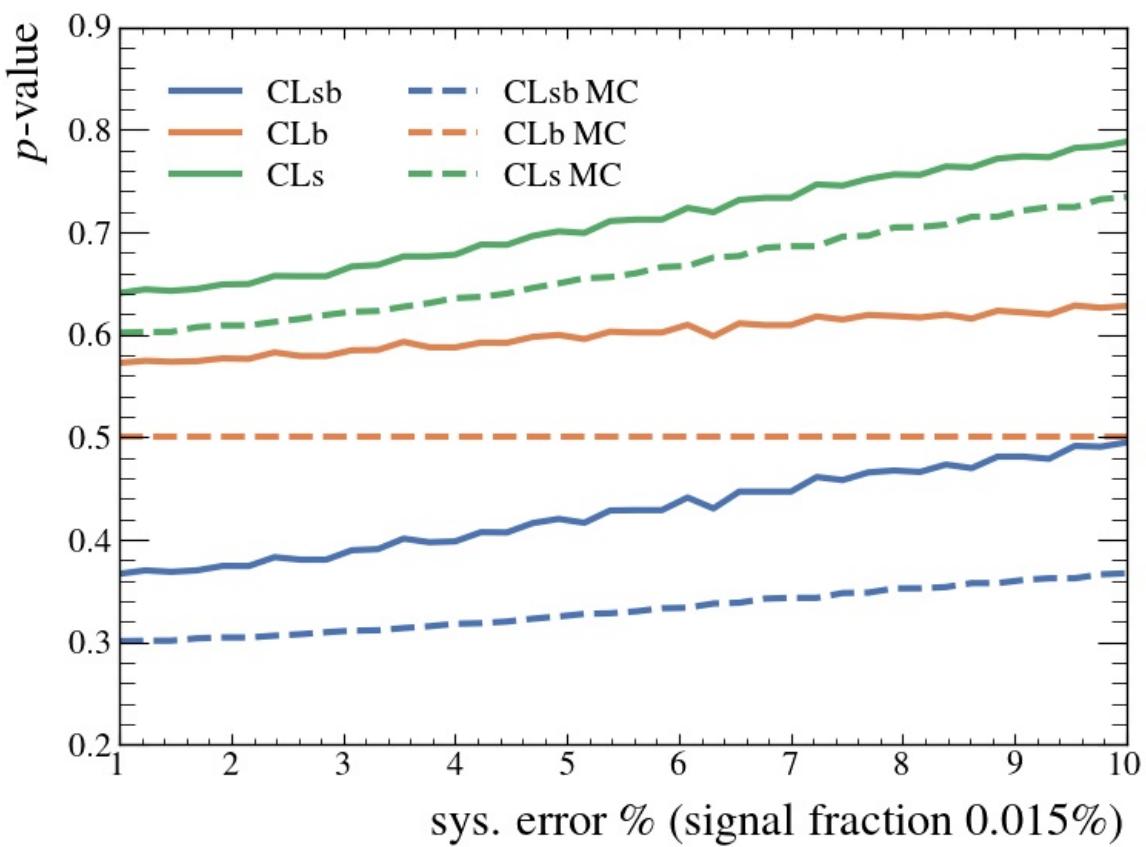
    plt.plot(mc_["bkg_err"] * 100, mc_["p_sb"], ls='--', c='C0', lw=3)
    plt.plot(mc_["bkg_err"] * 100, mc_["p_b"], ls='--', c='C1', lw=3)
    plt.plot(mc_["bkg_err"] * 100, mc_["p_s"], ls='--', c='C2', lw=3)

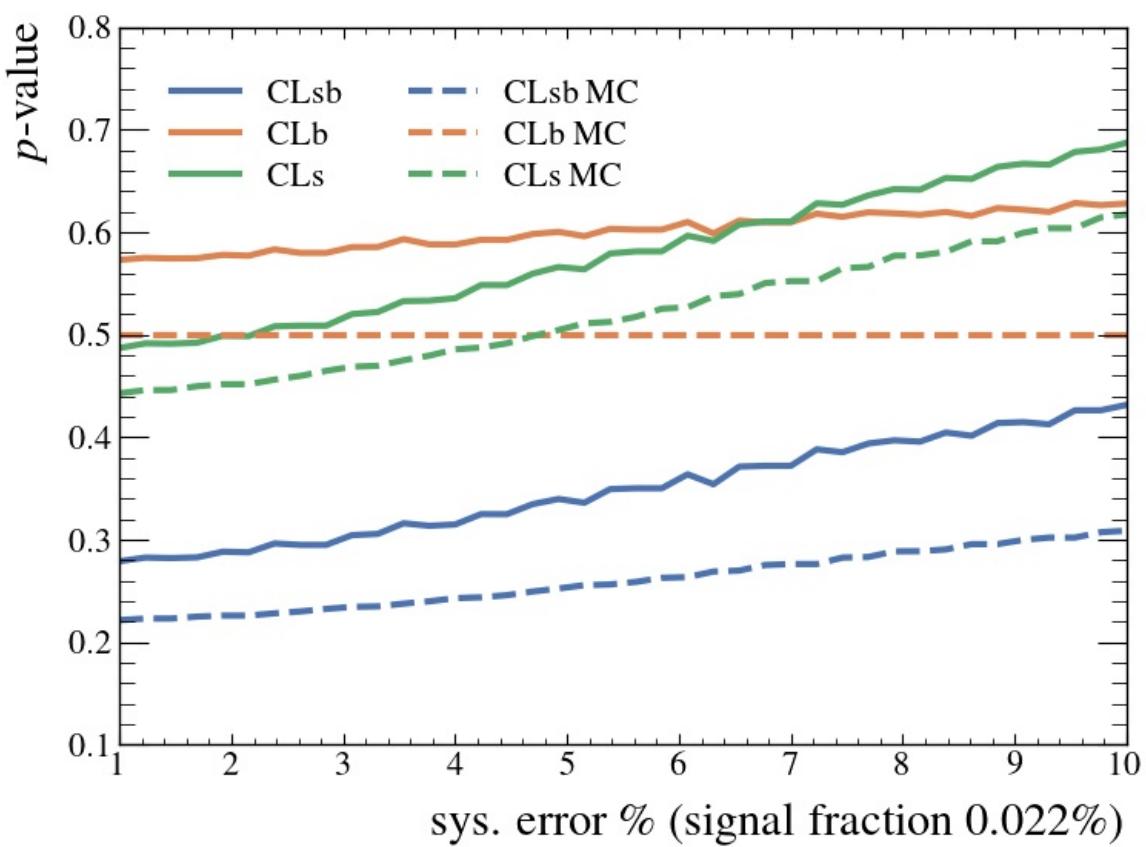
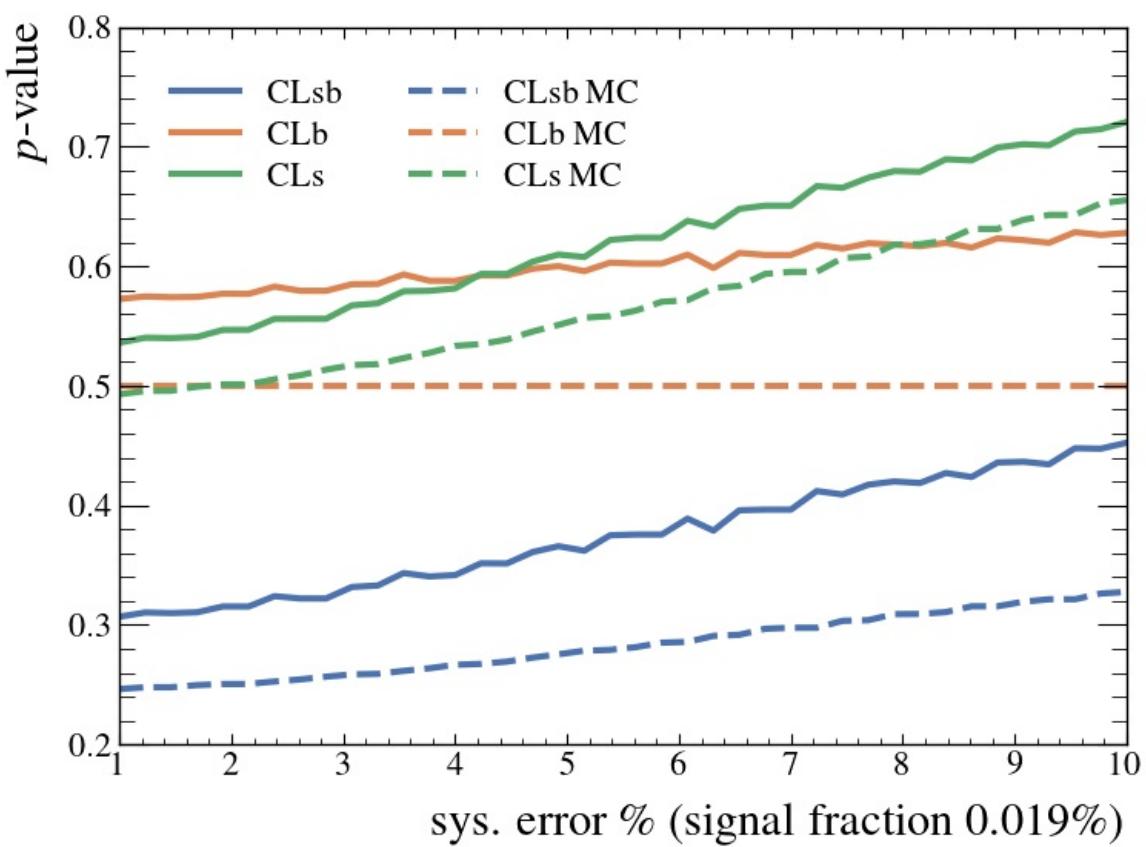
    plt.legend(["CLsb", "CLb", "CLs", "CLsb MC", "CLb MC", "CLs MC"], ncol=2)

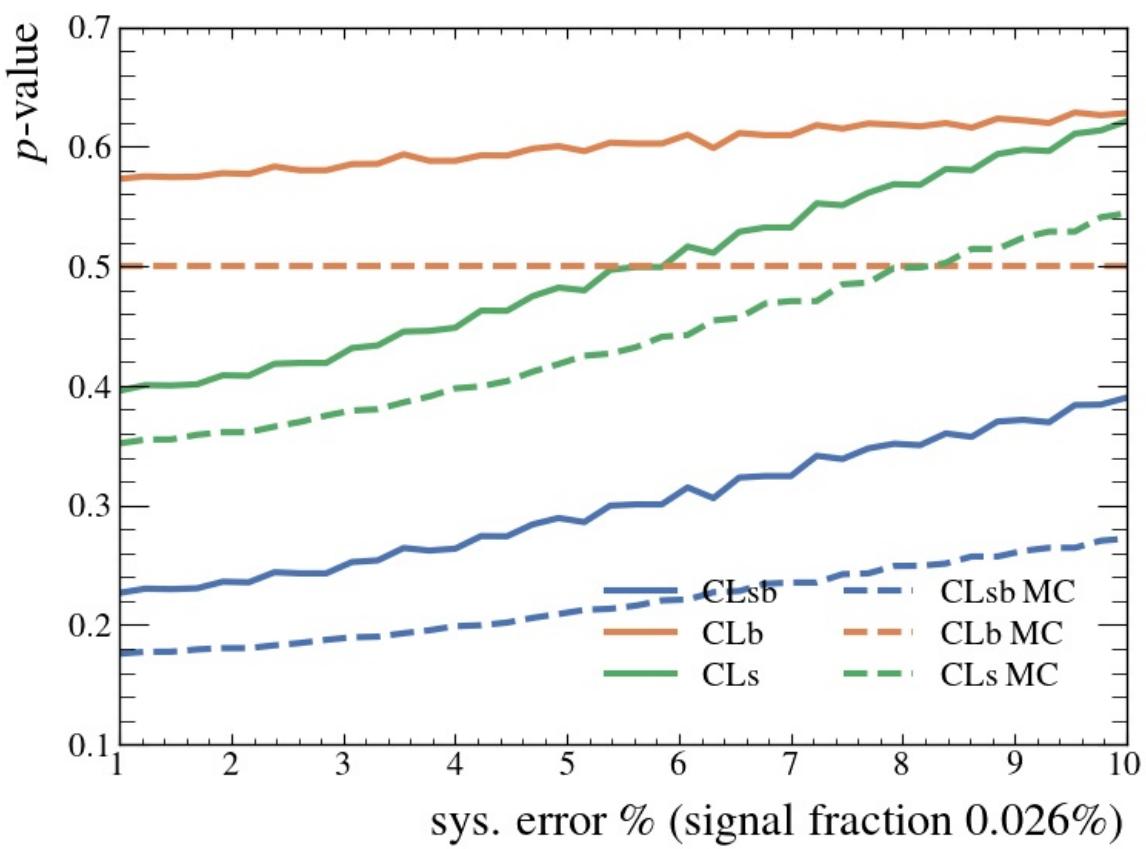
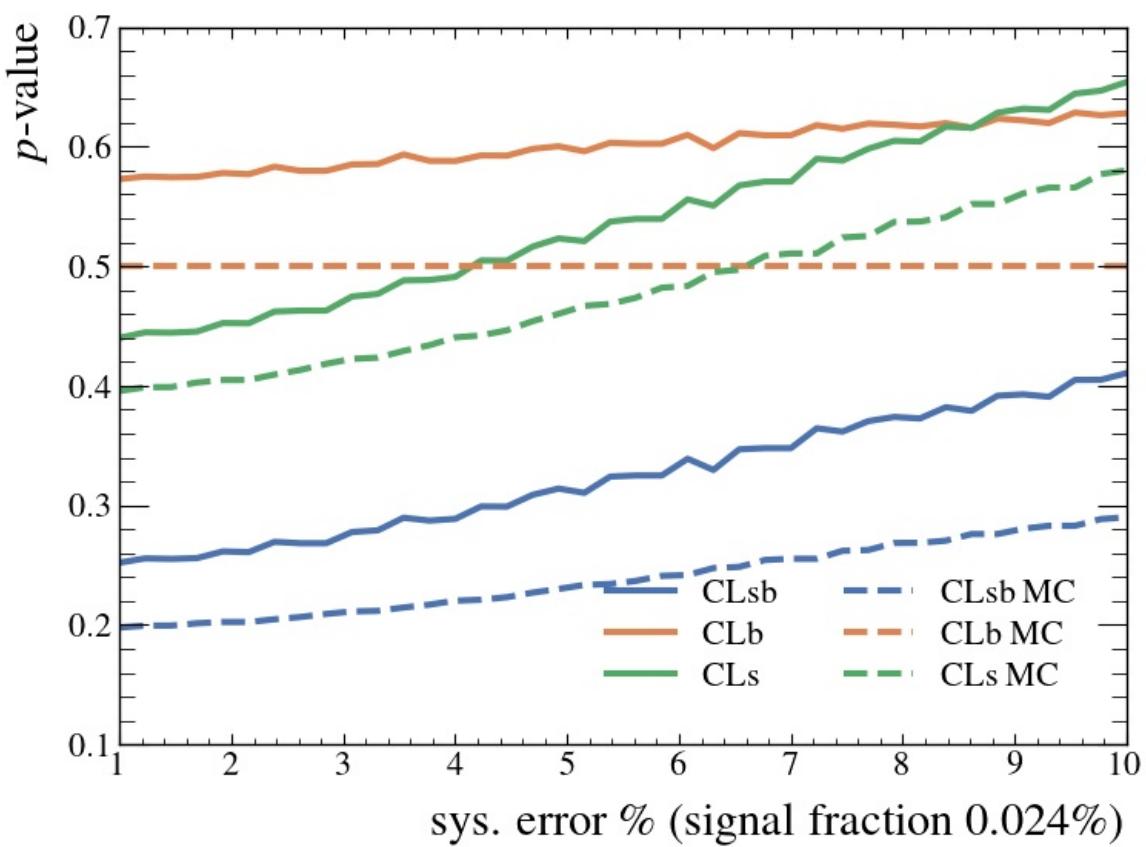
    plt.xlabel("sys. error \% (signal fraction {:.3f}\%)".format(f), fontsize=22)
    plt.ylabel("$p\$-value", fontsize=22)
    plt.tight_layout()

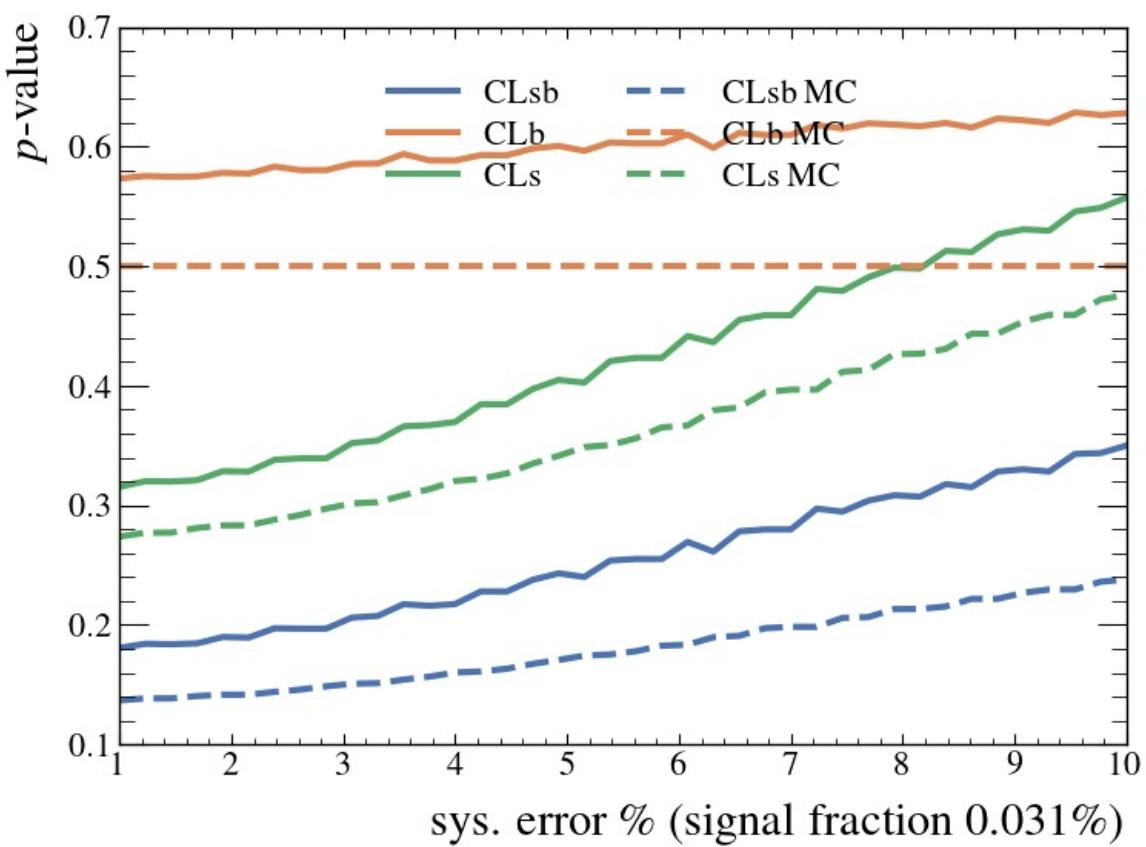
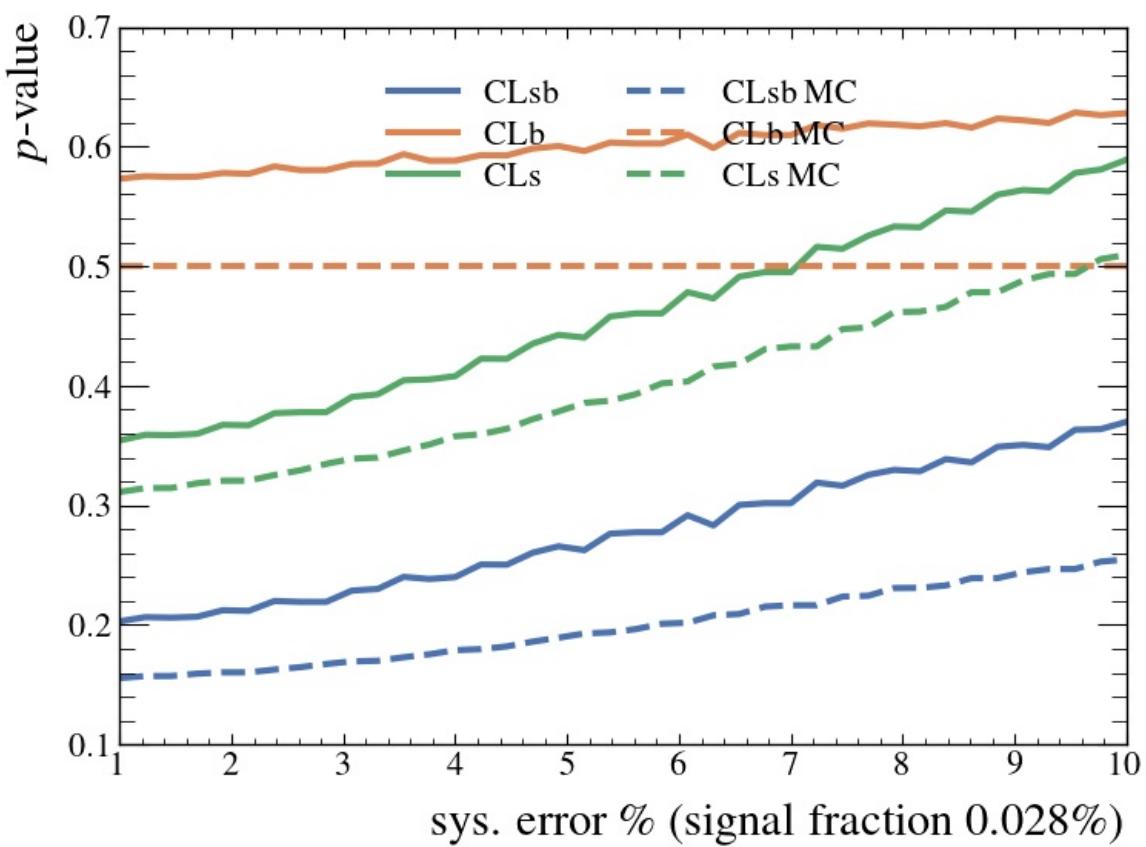
    plt.savefig(saved + f"CLs_q0_mu0_bkg_errs_{i}.pdf")
    plt.show()
```

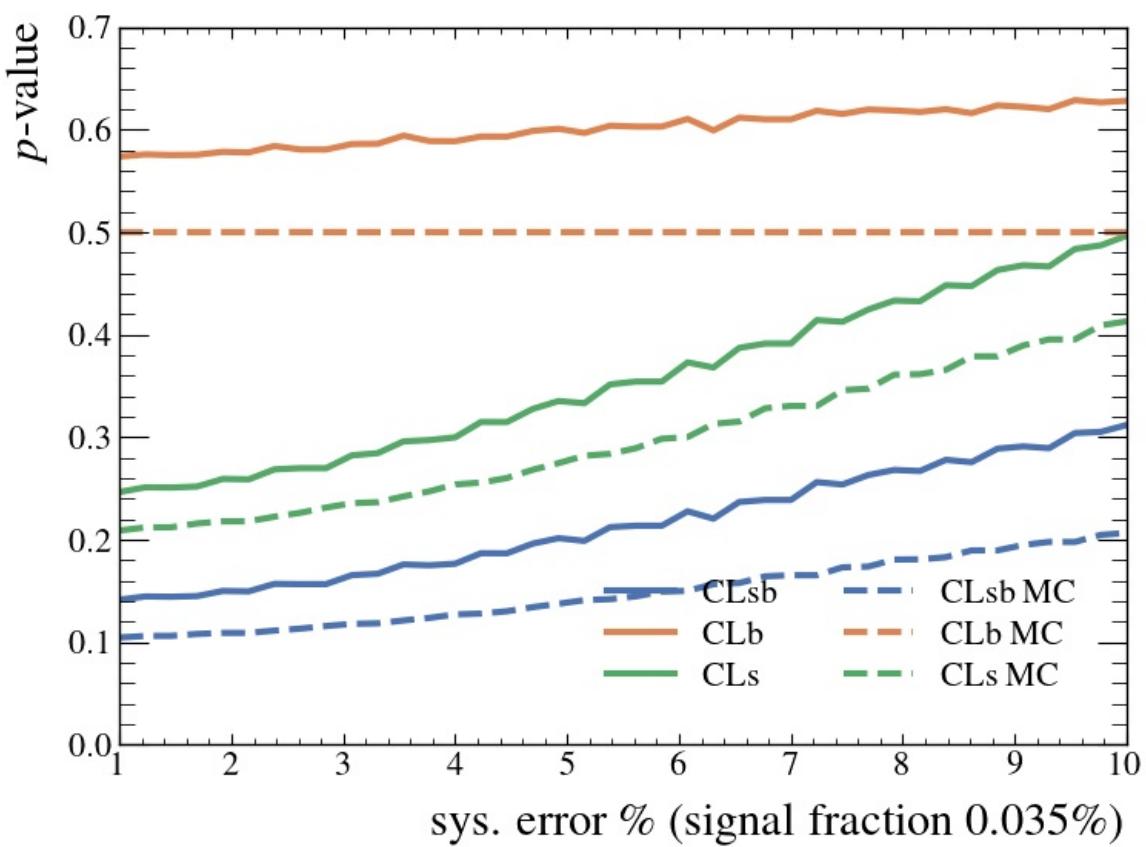
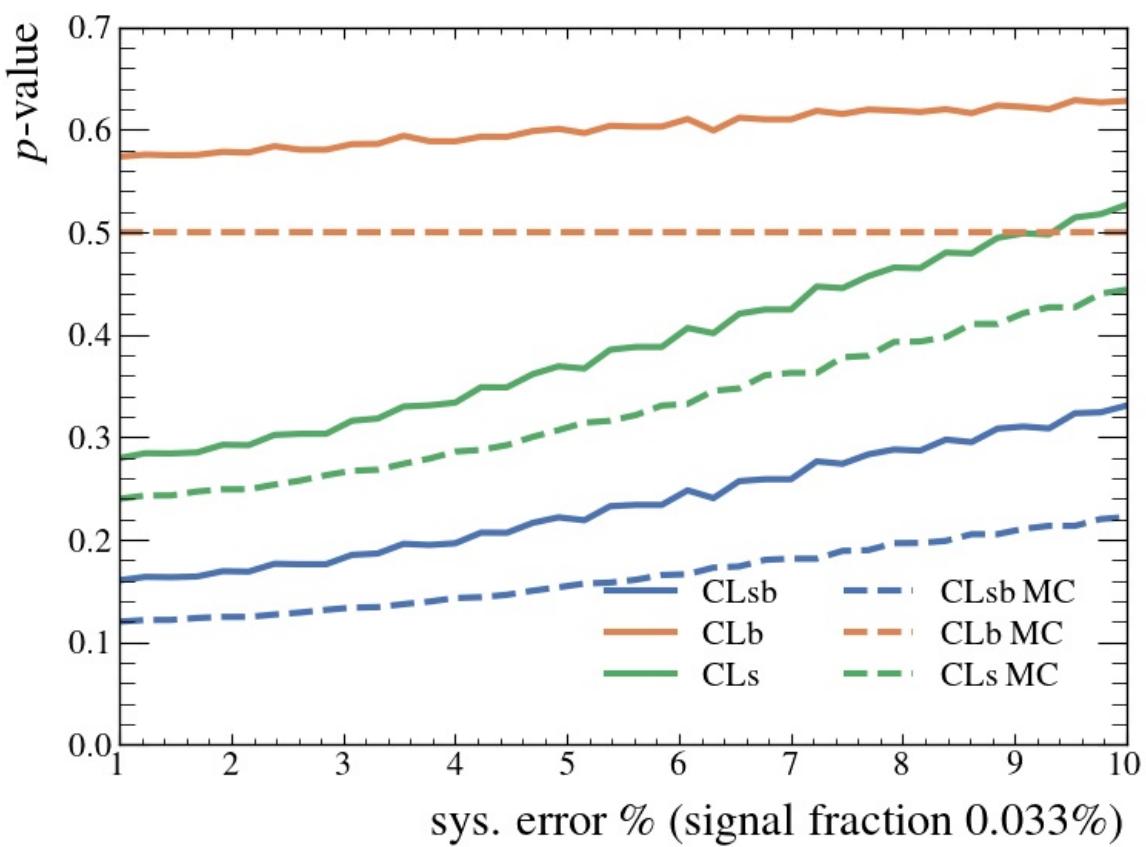


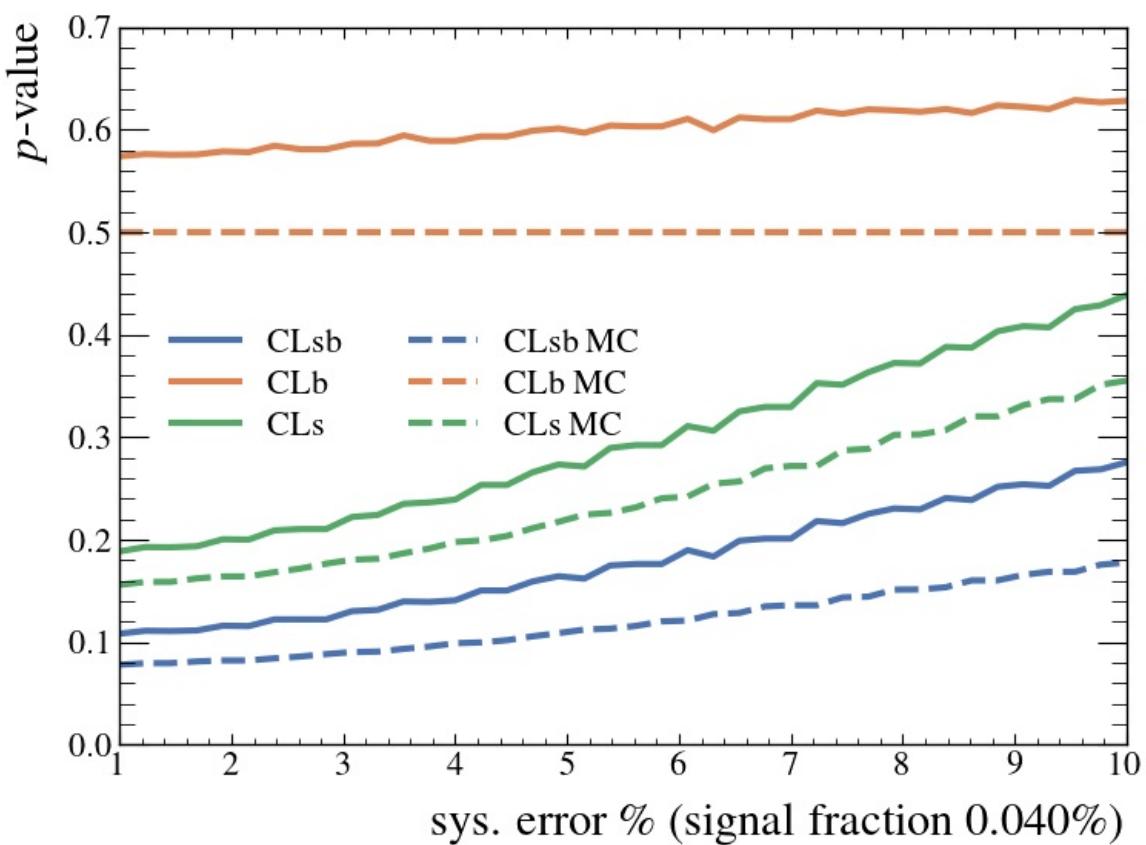
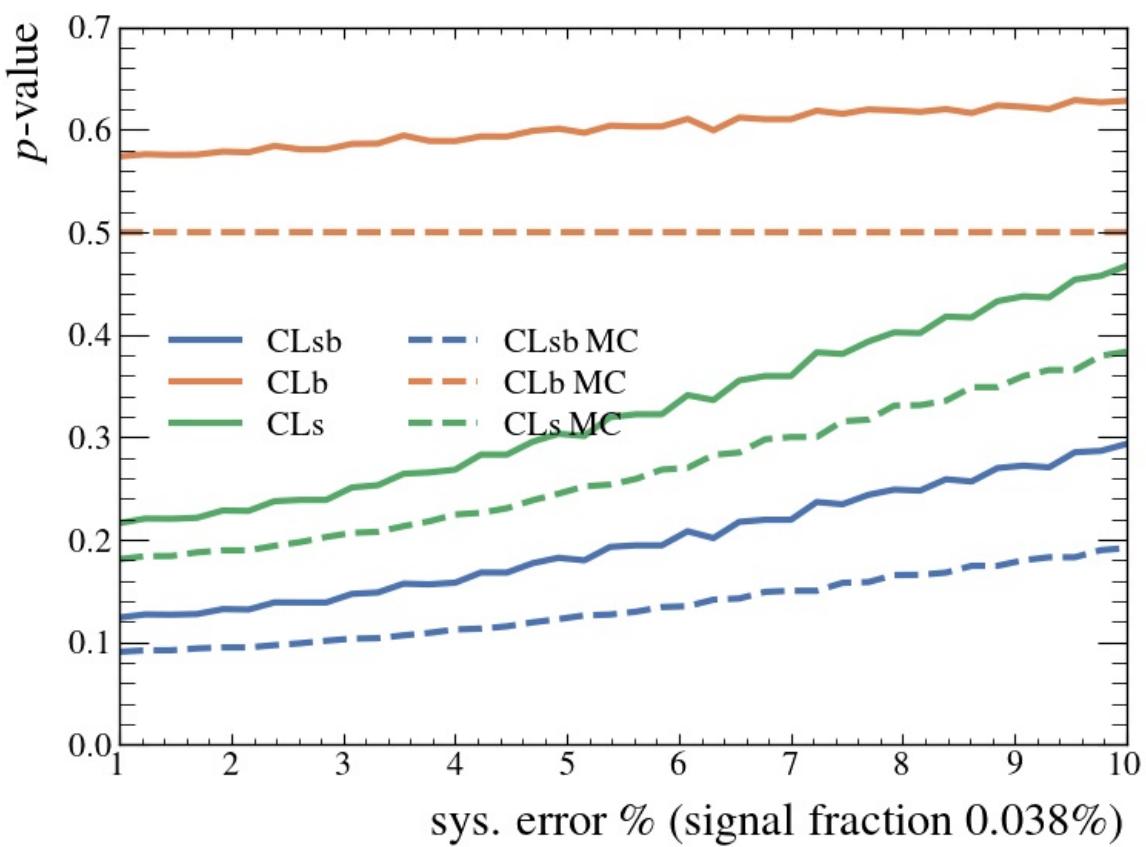


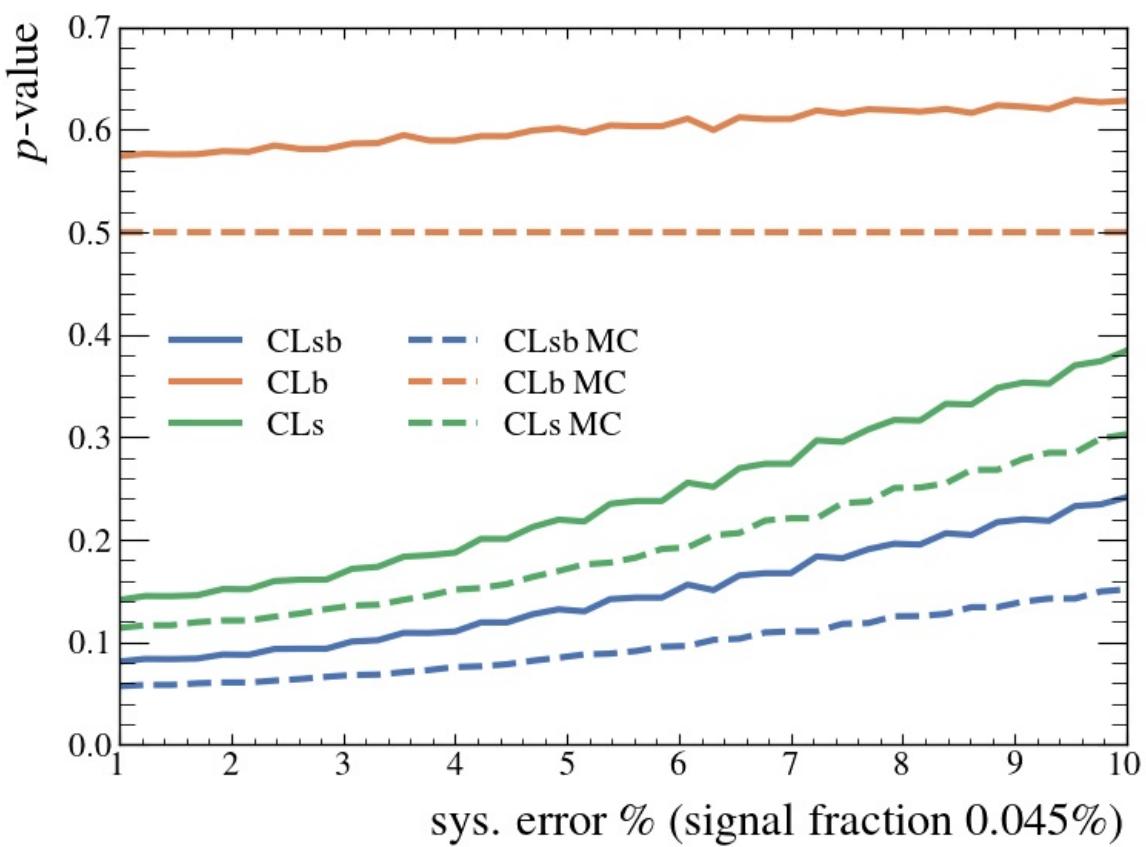
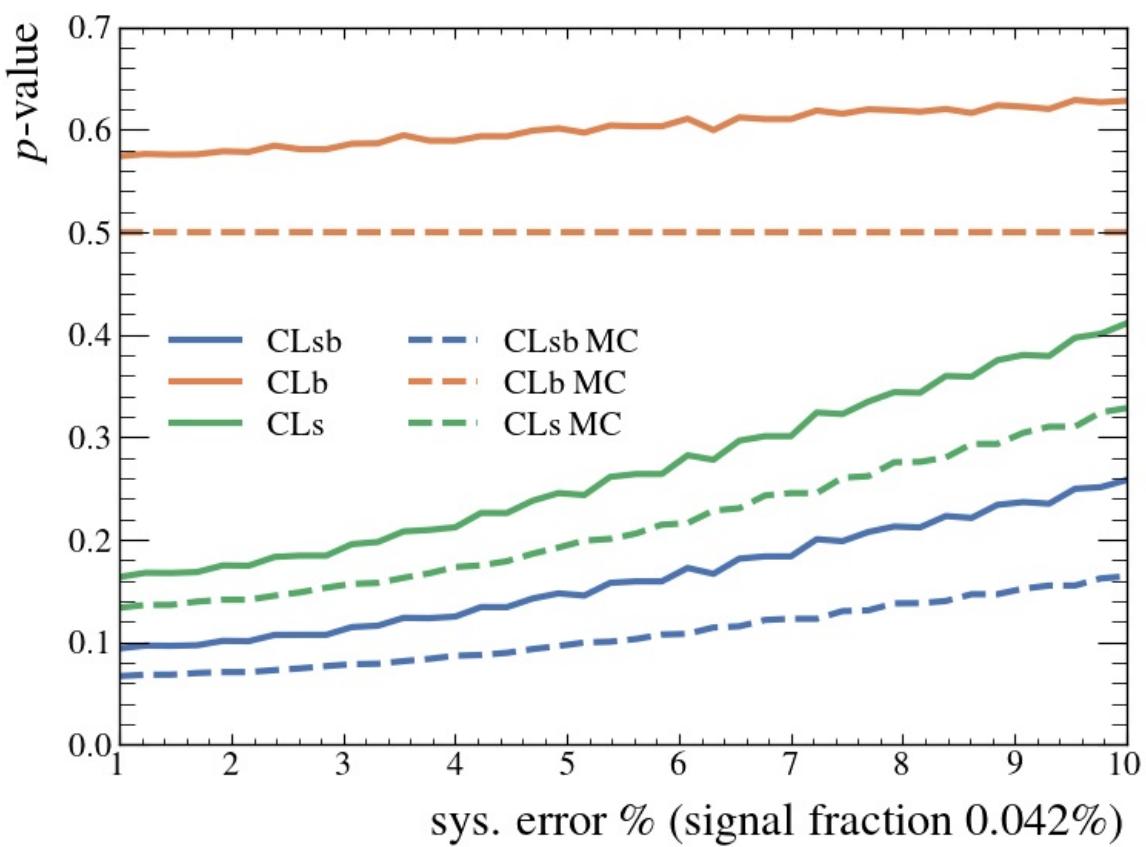


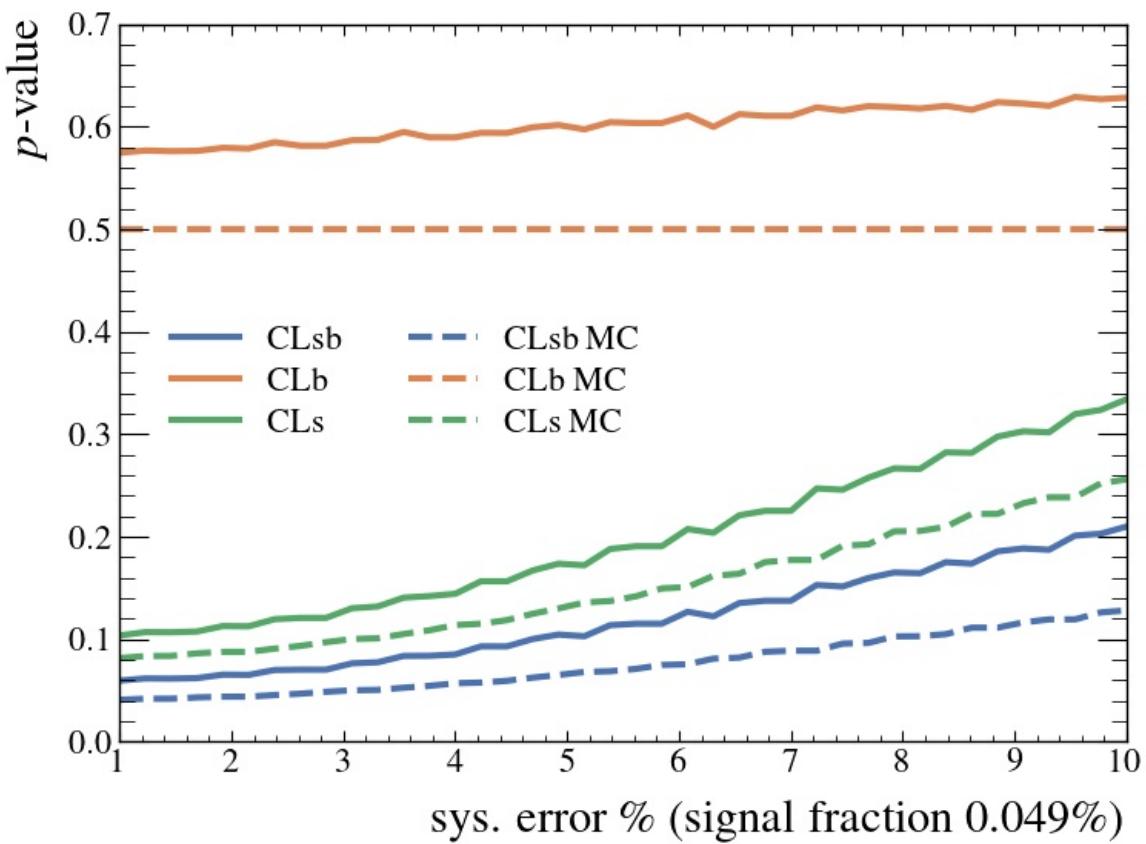
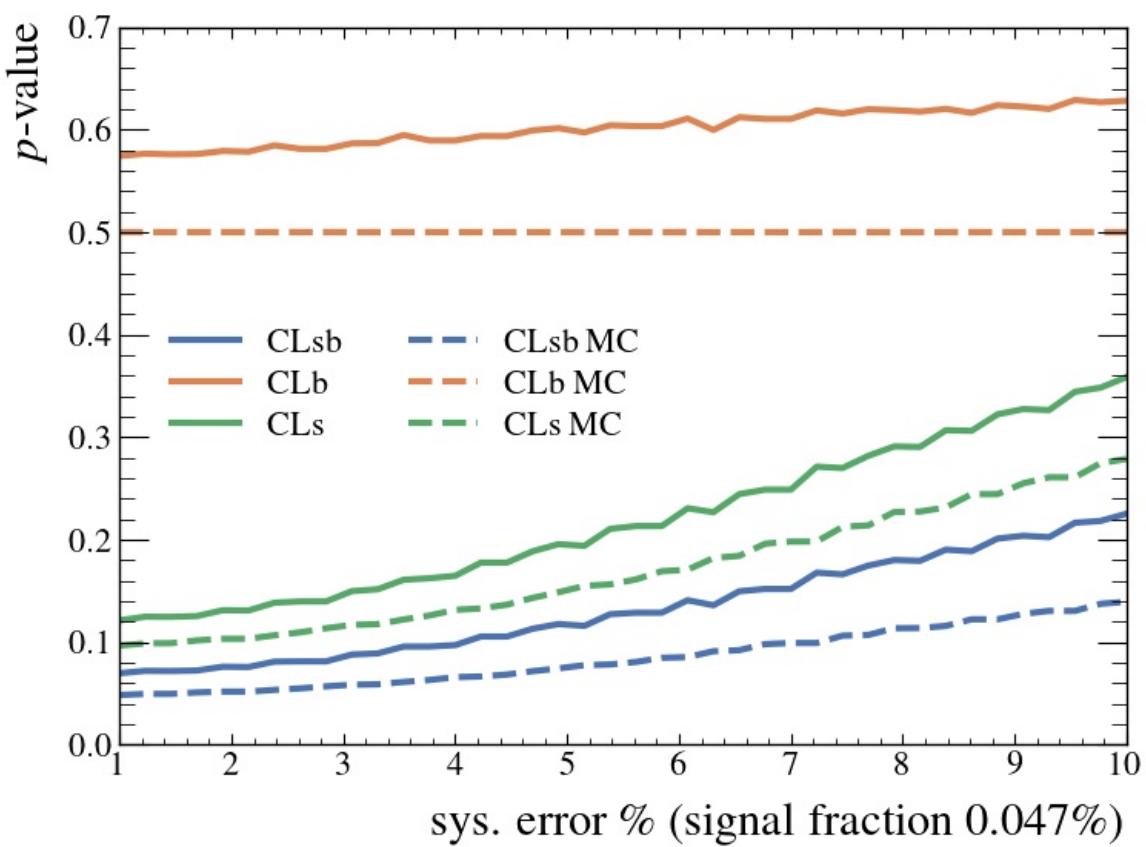


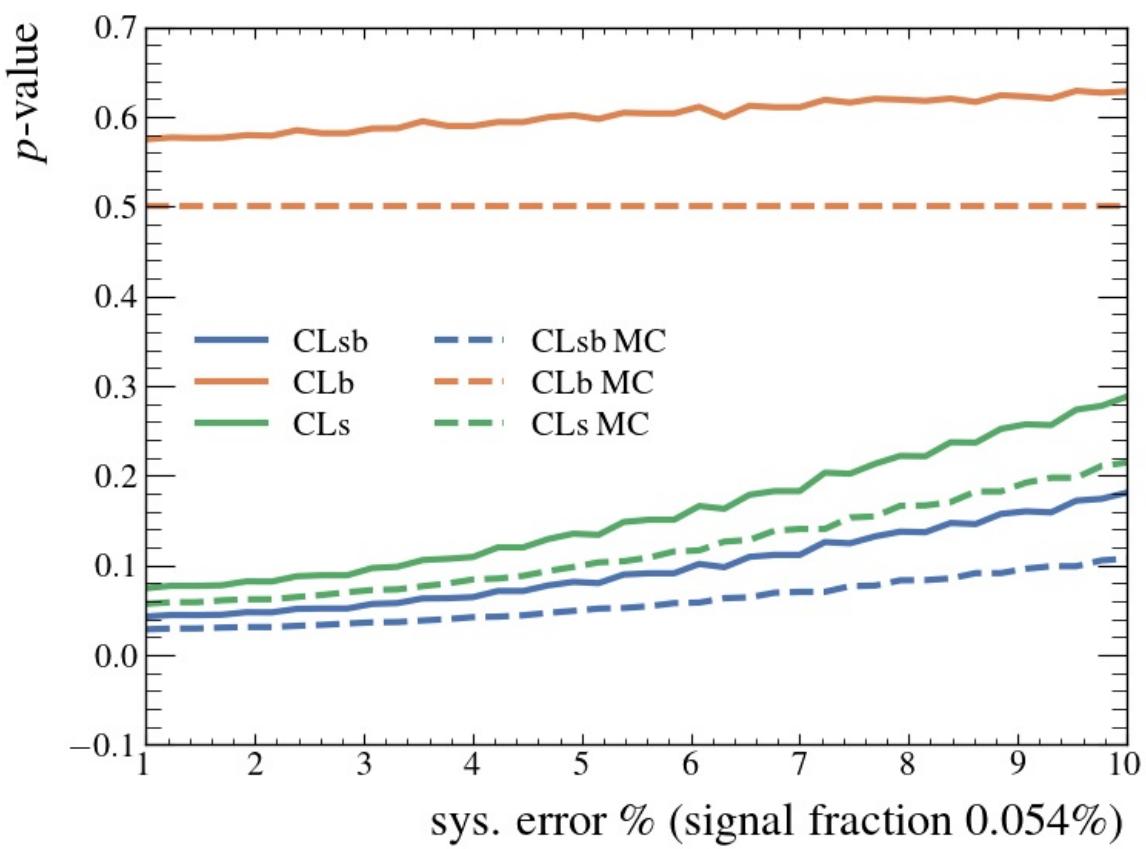
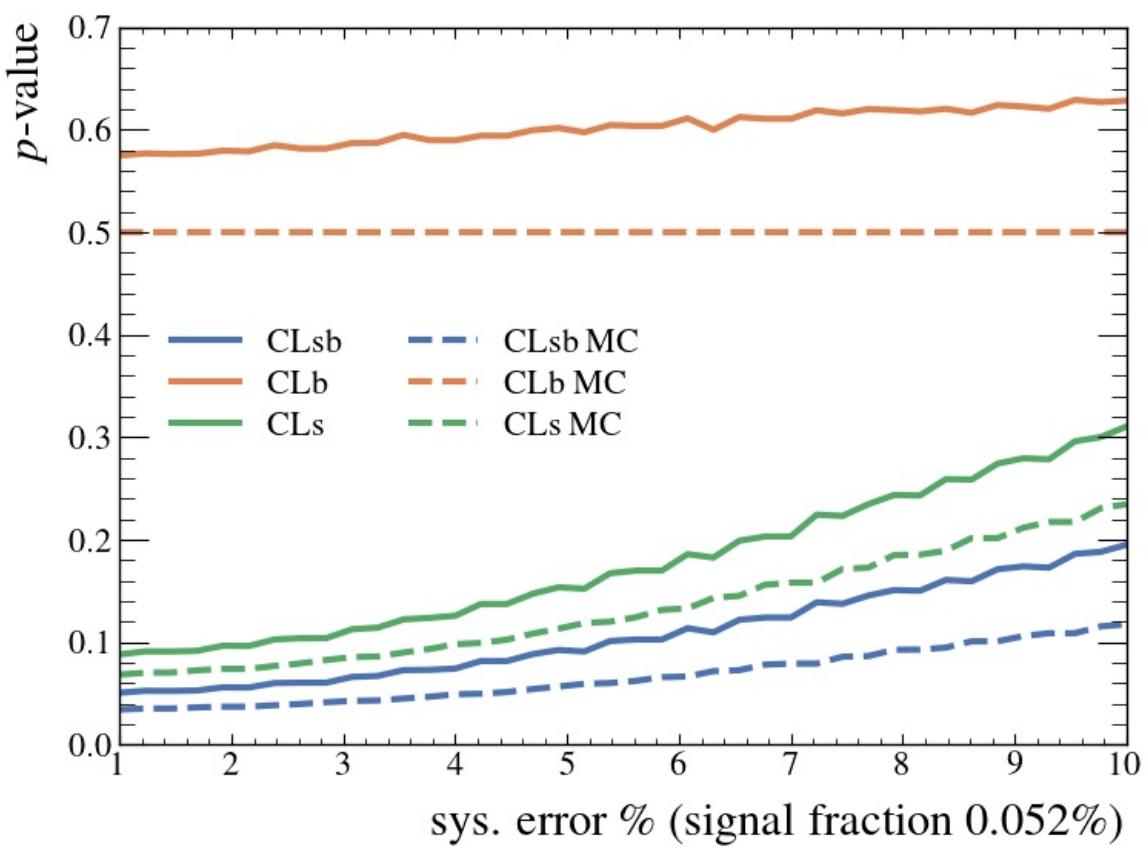


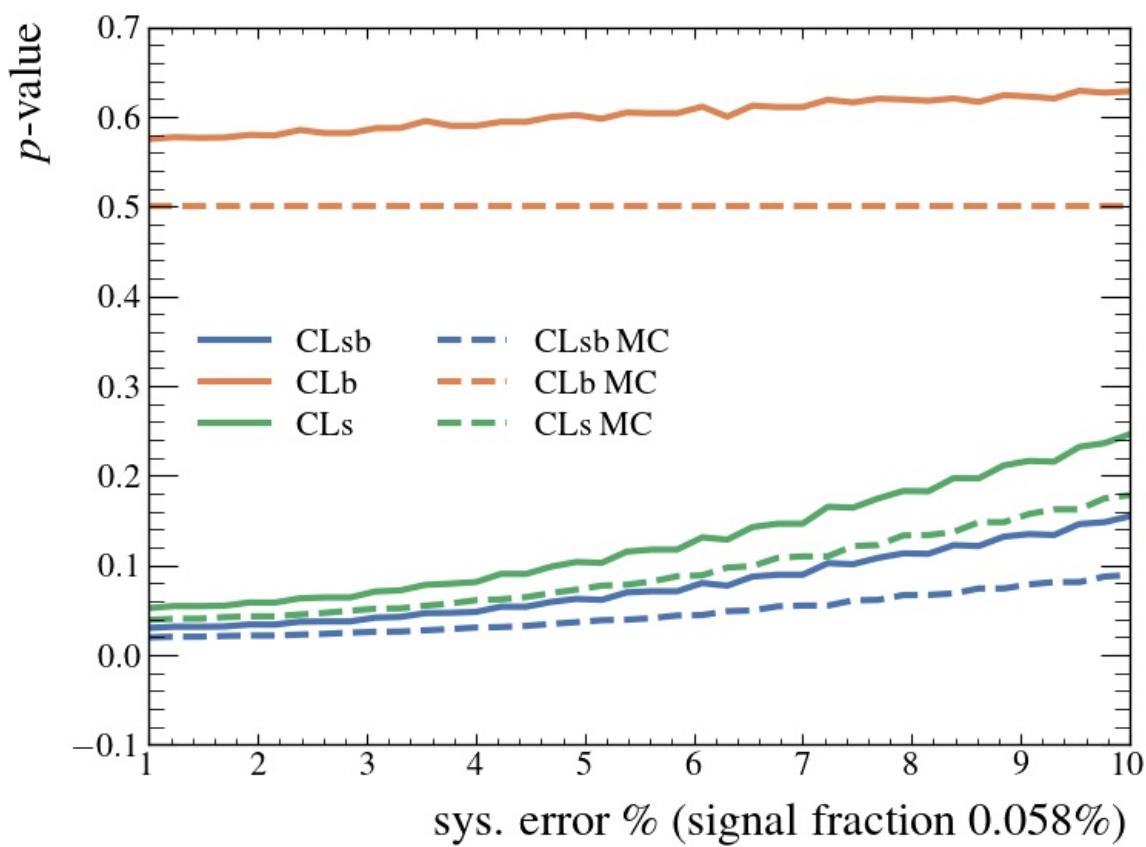
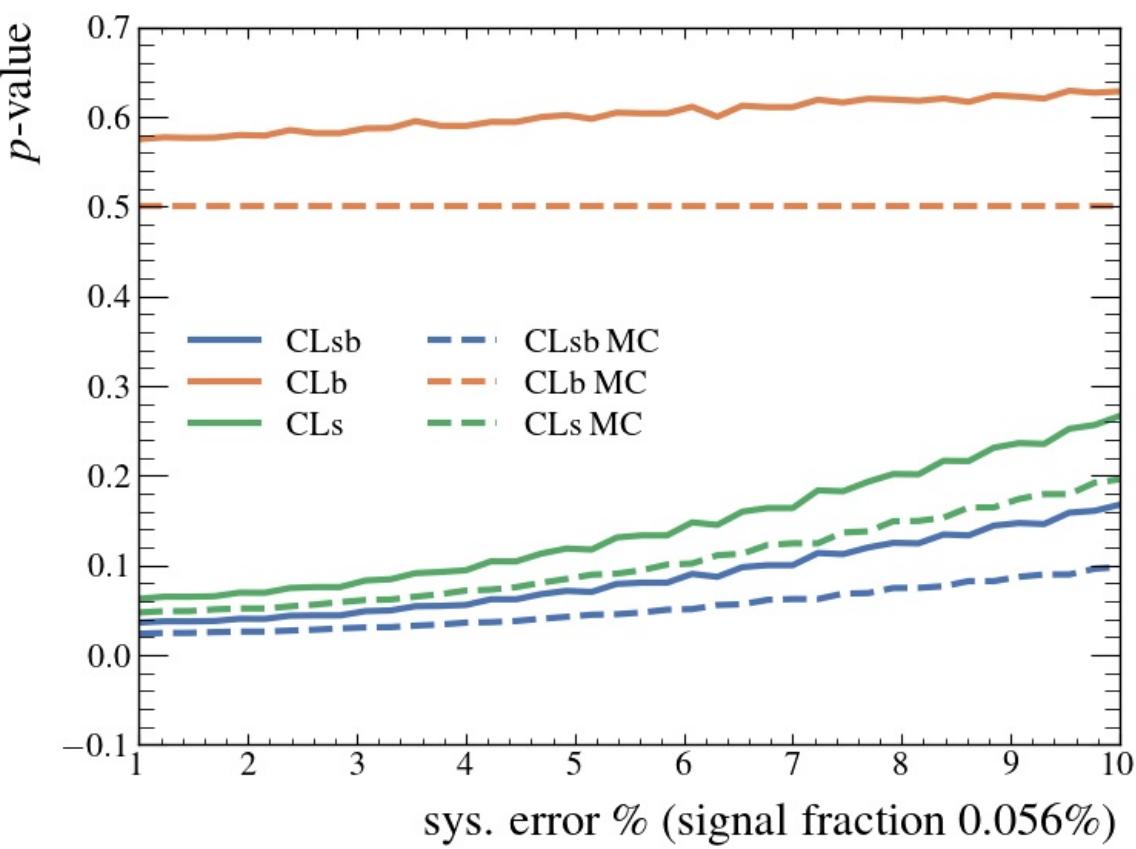


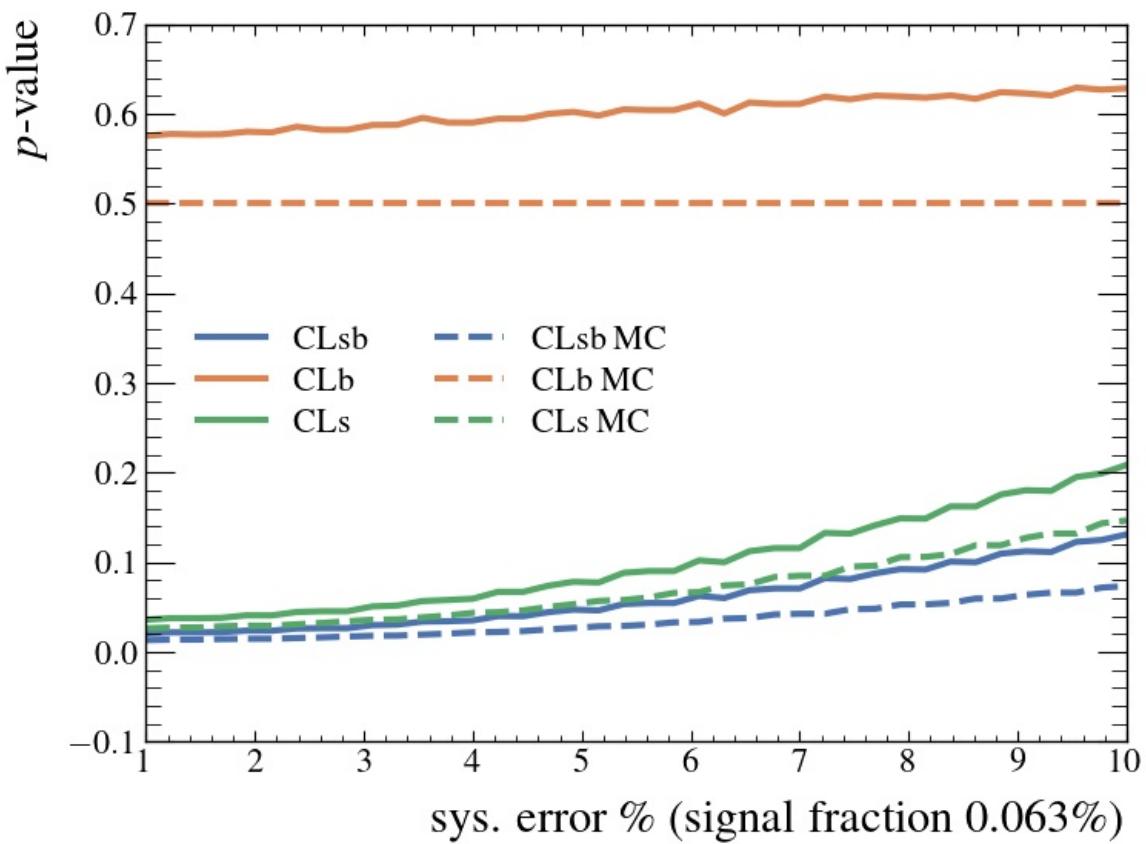
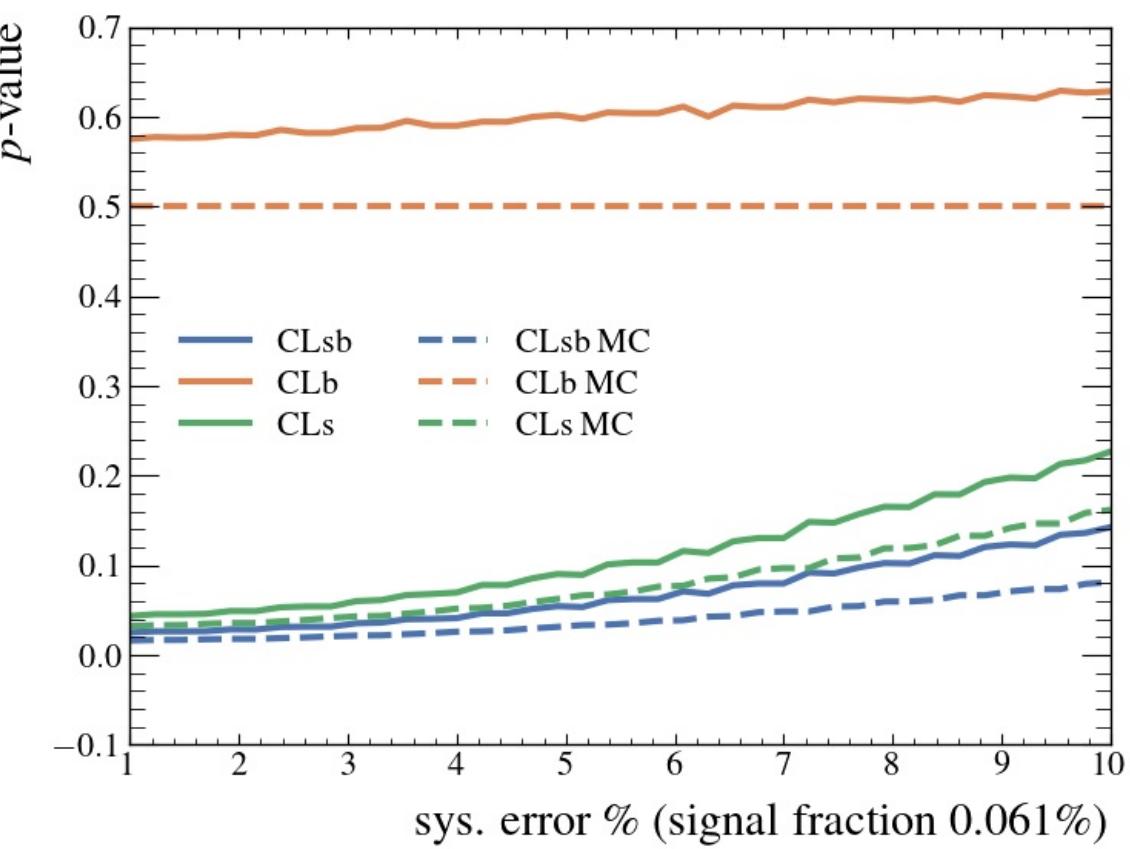


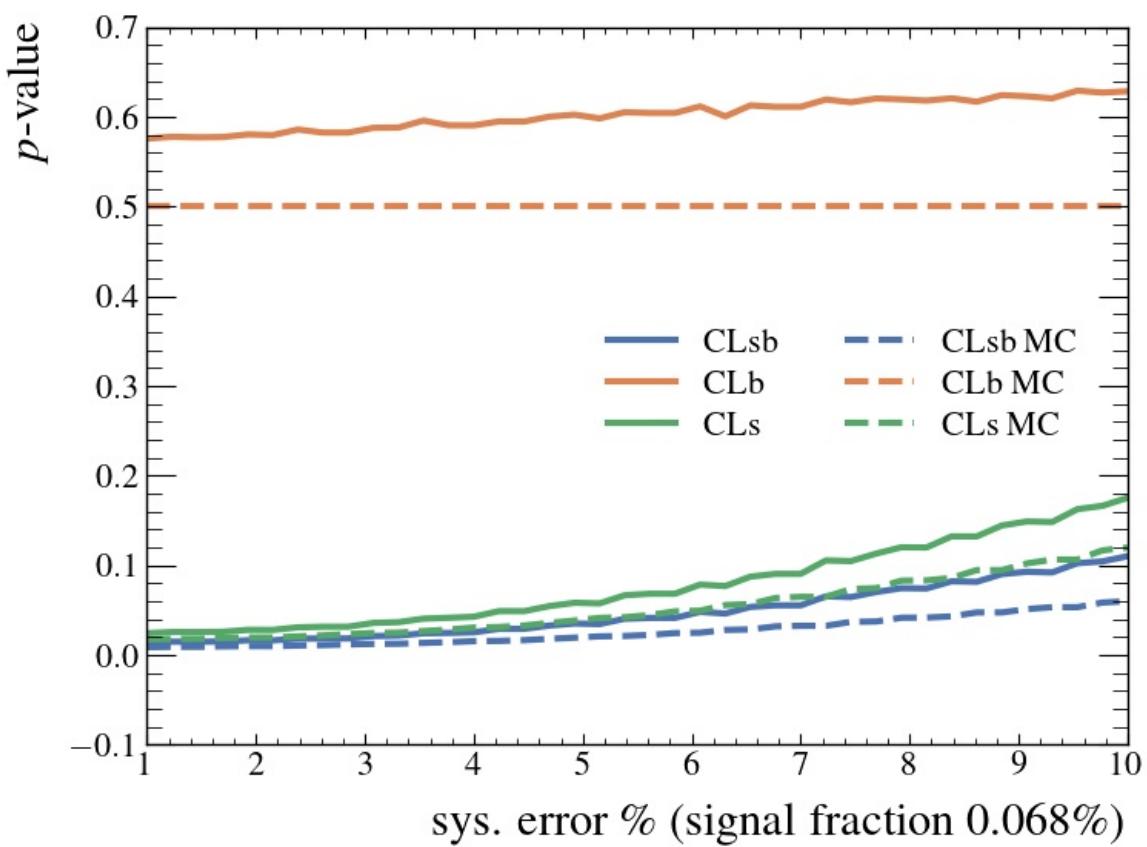
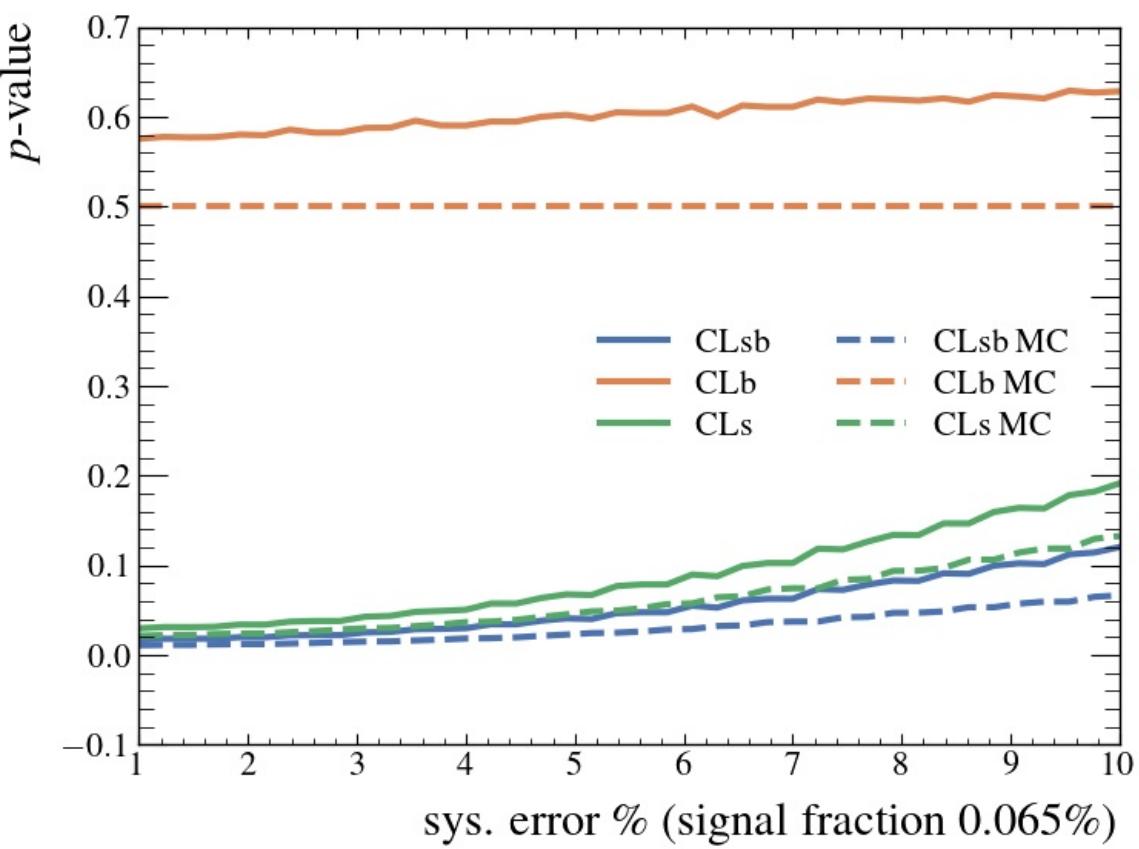


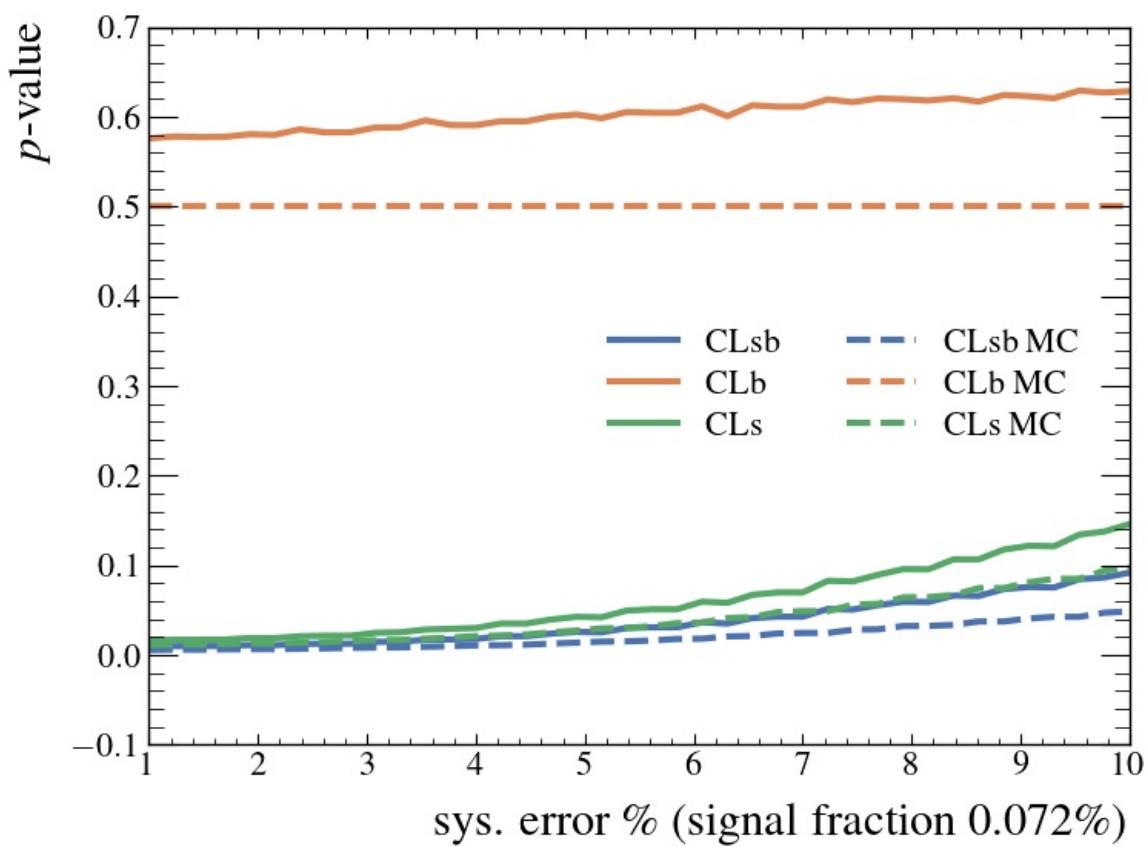
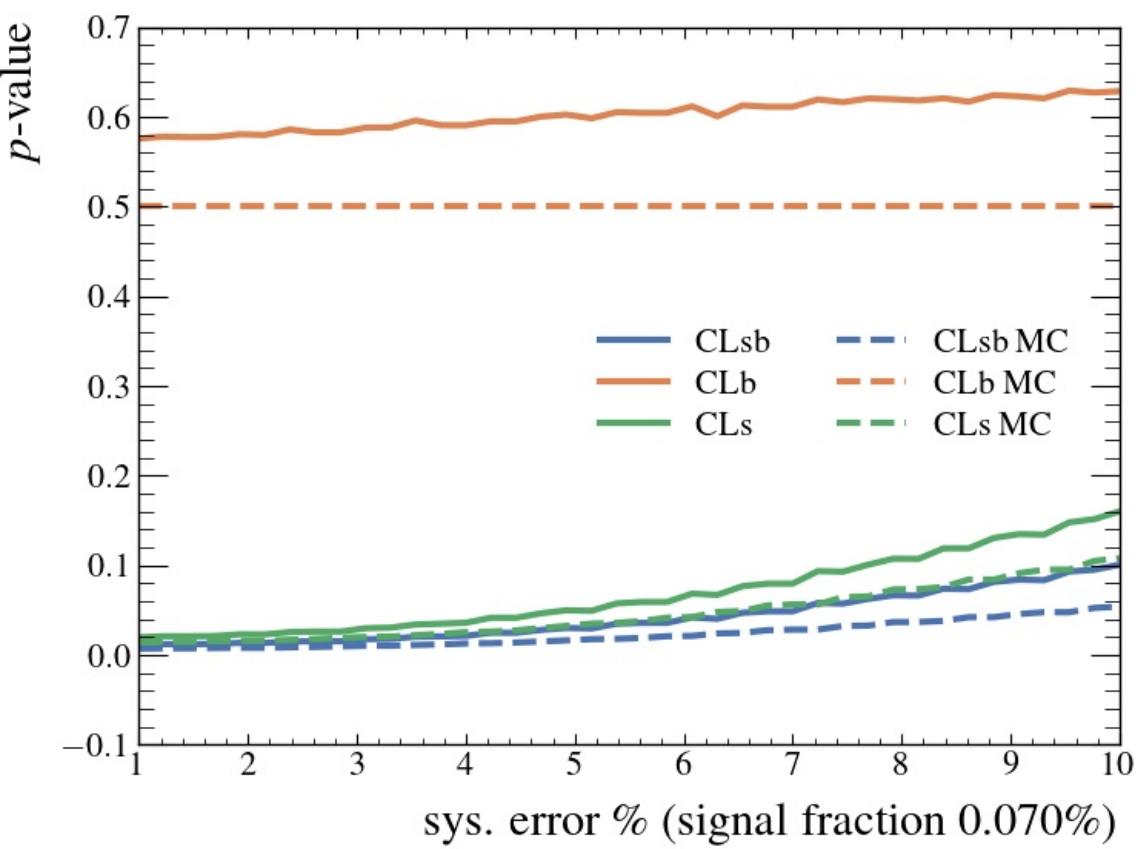


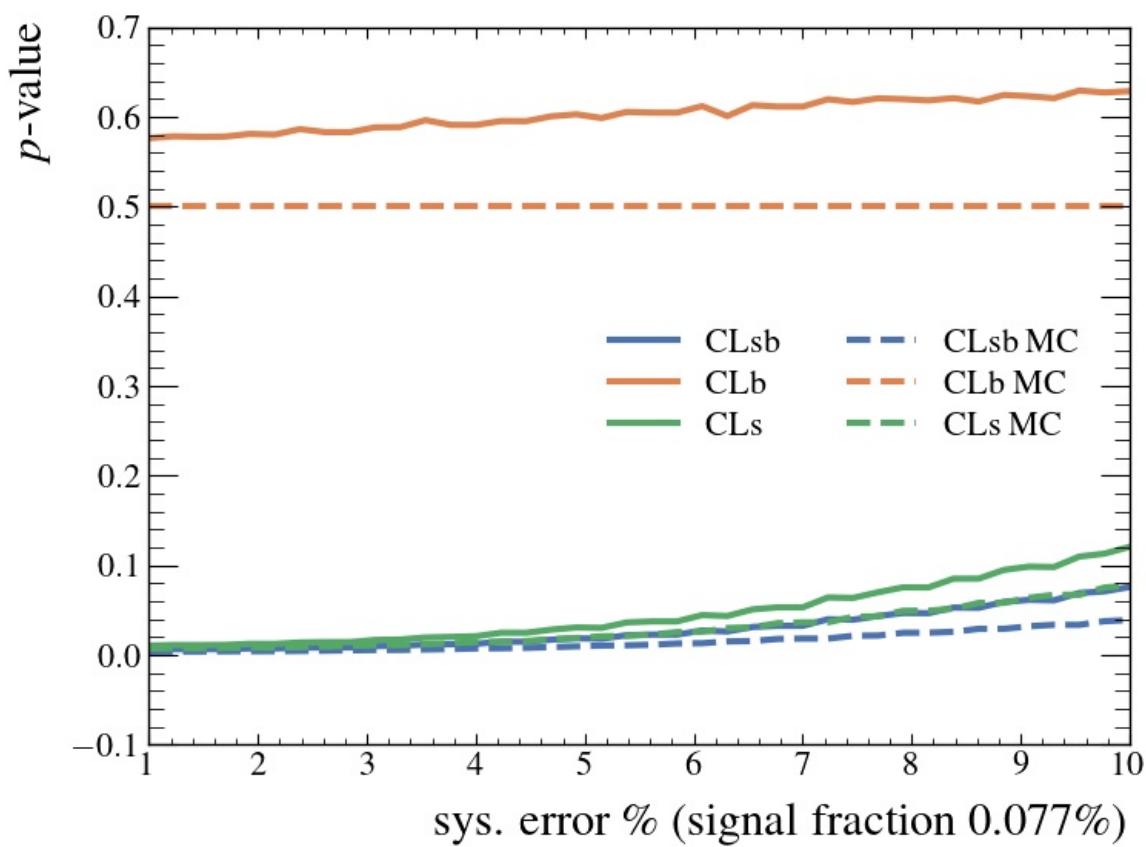
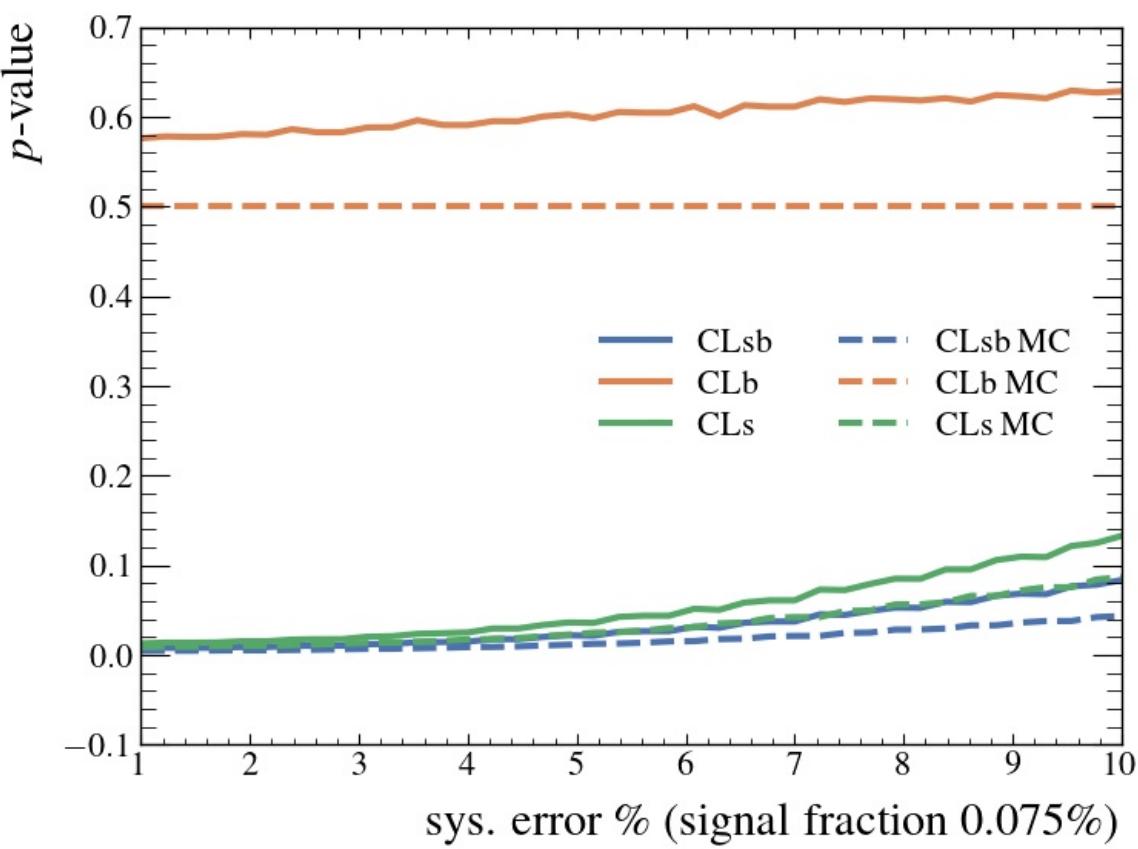


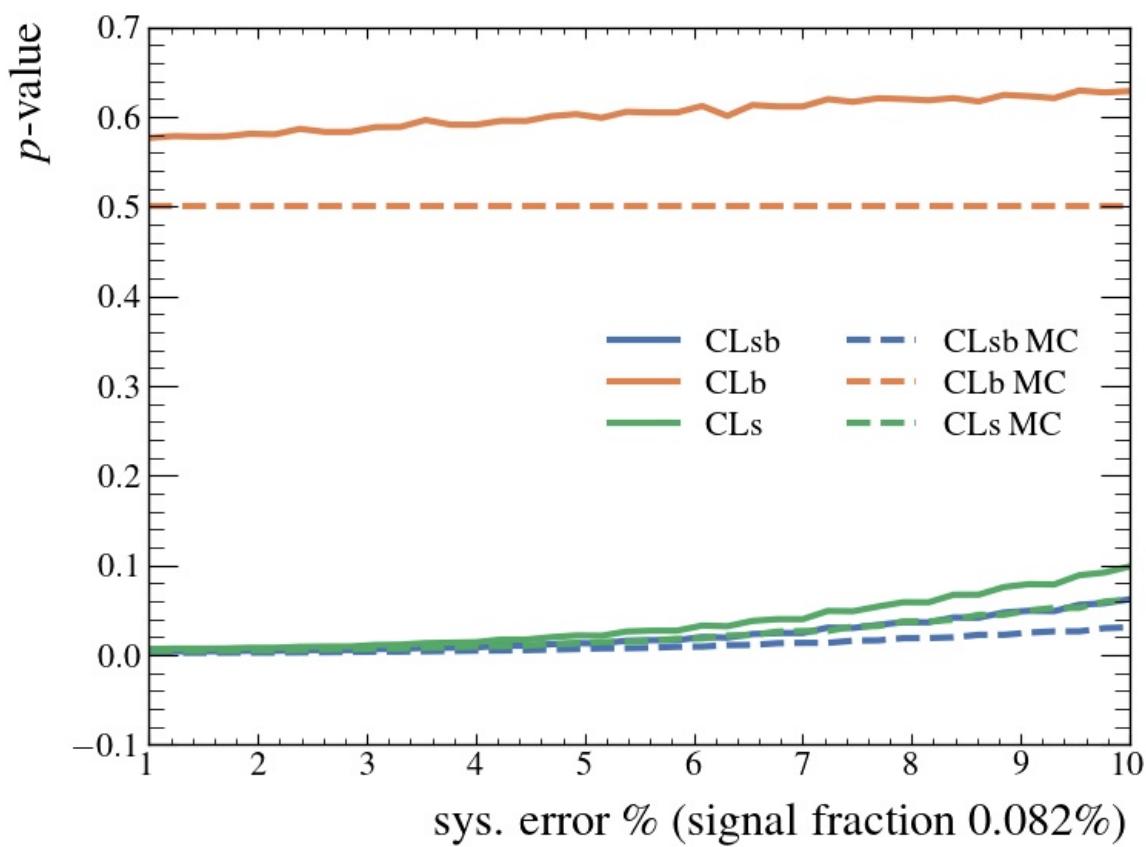
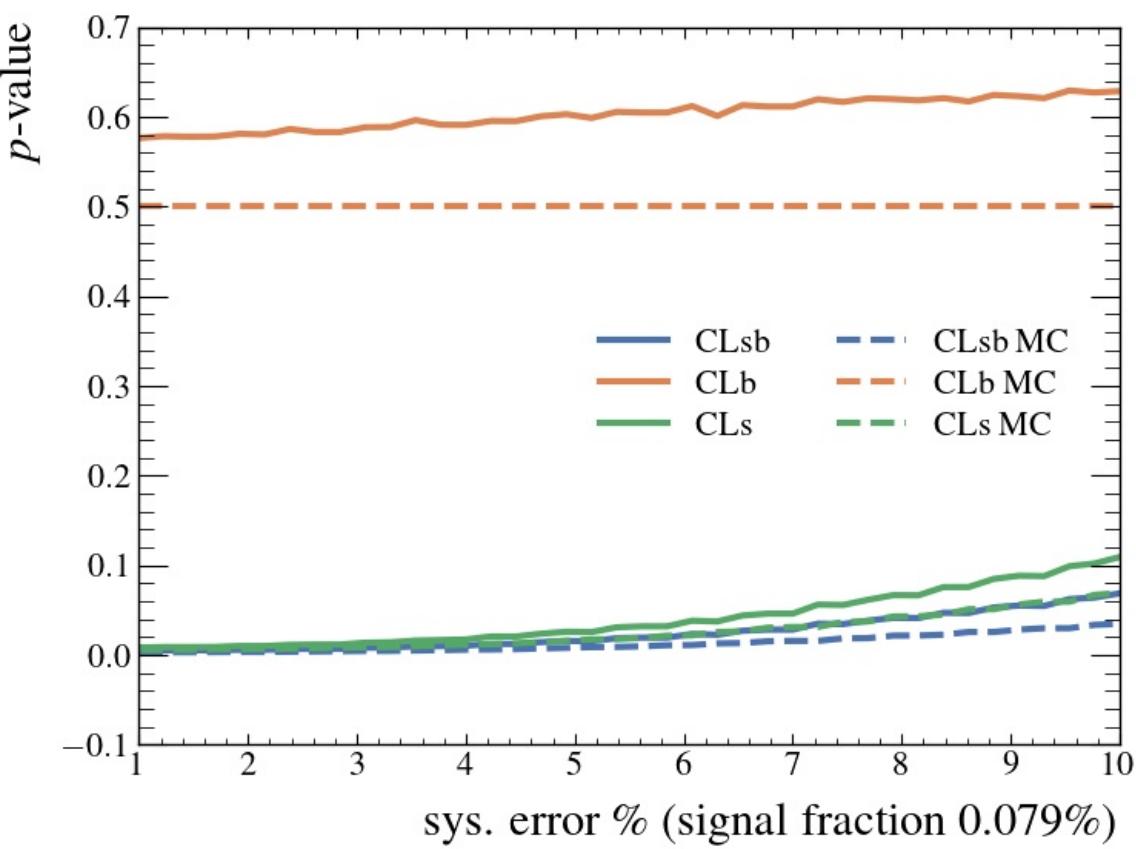


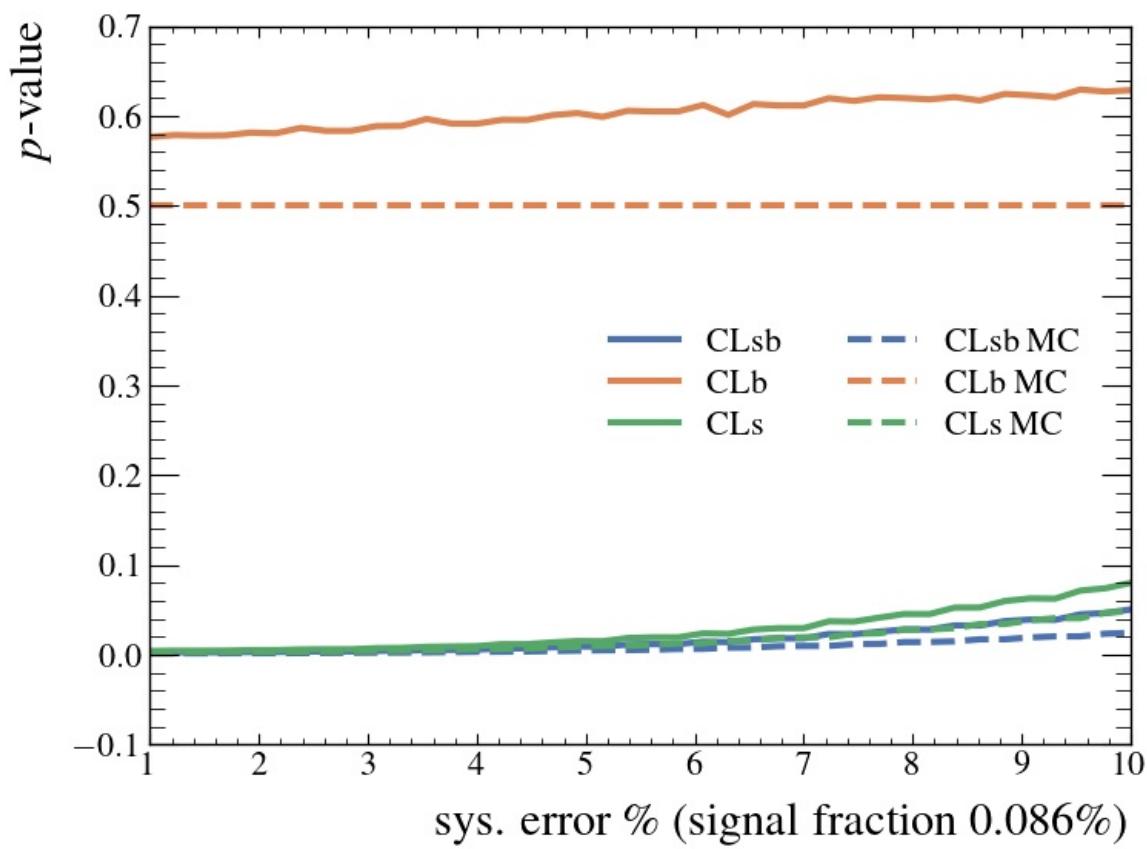
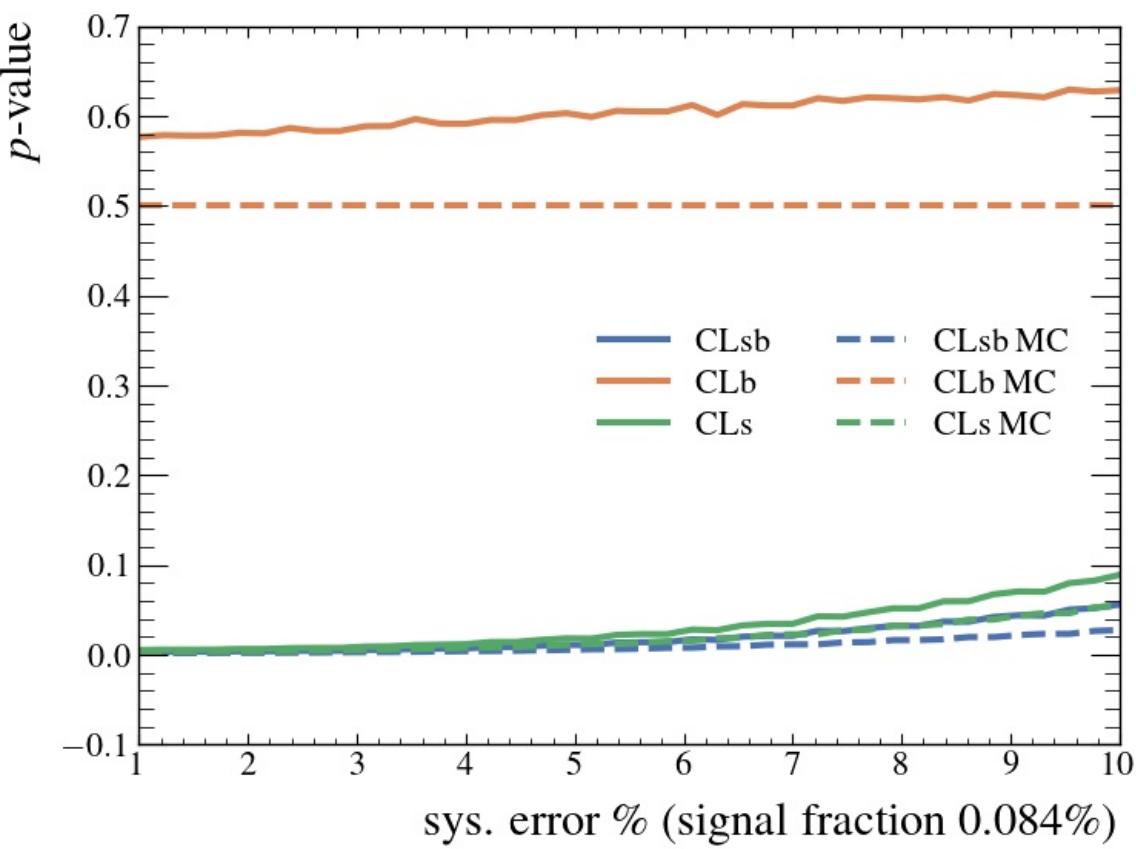


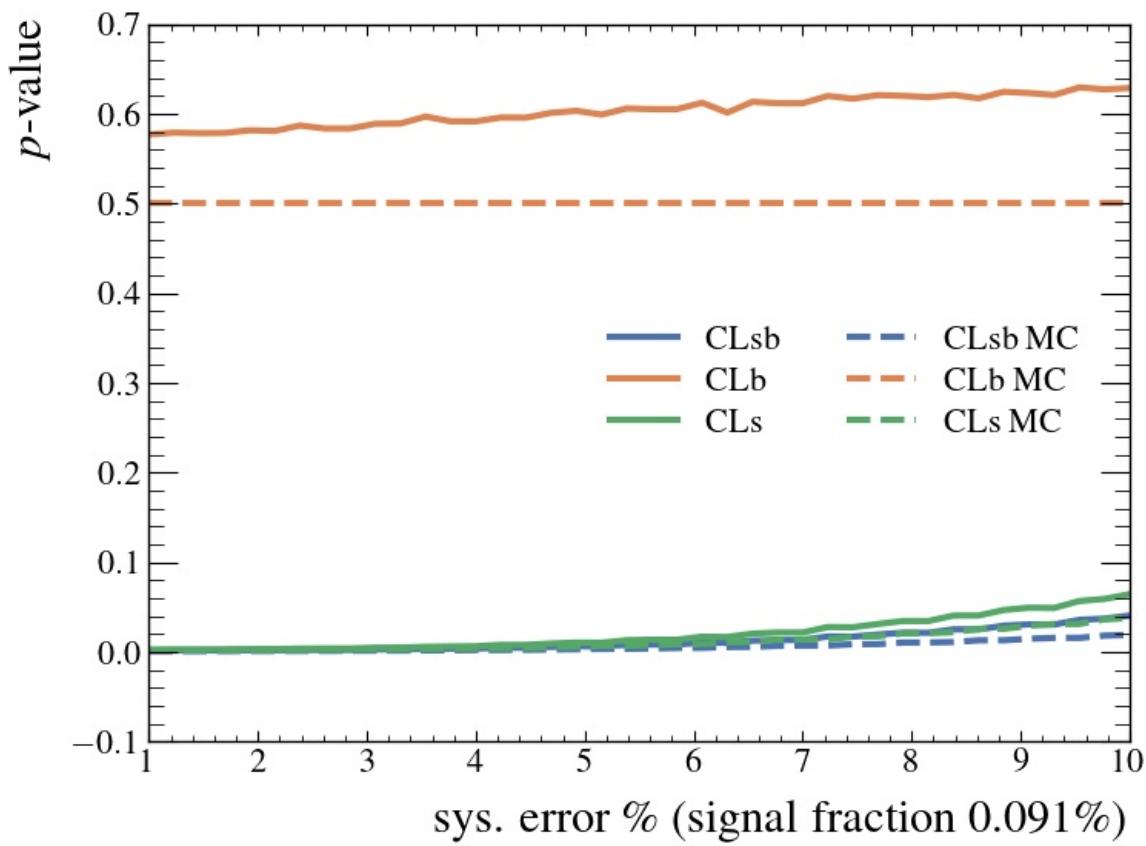
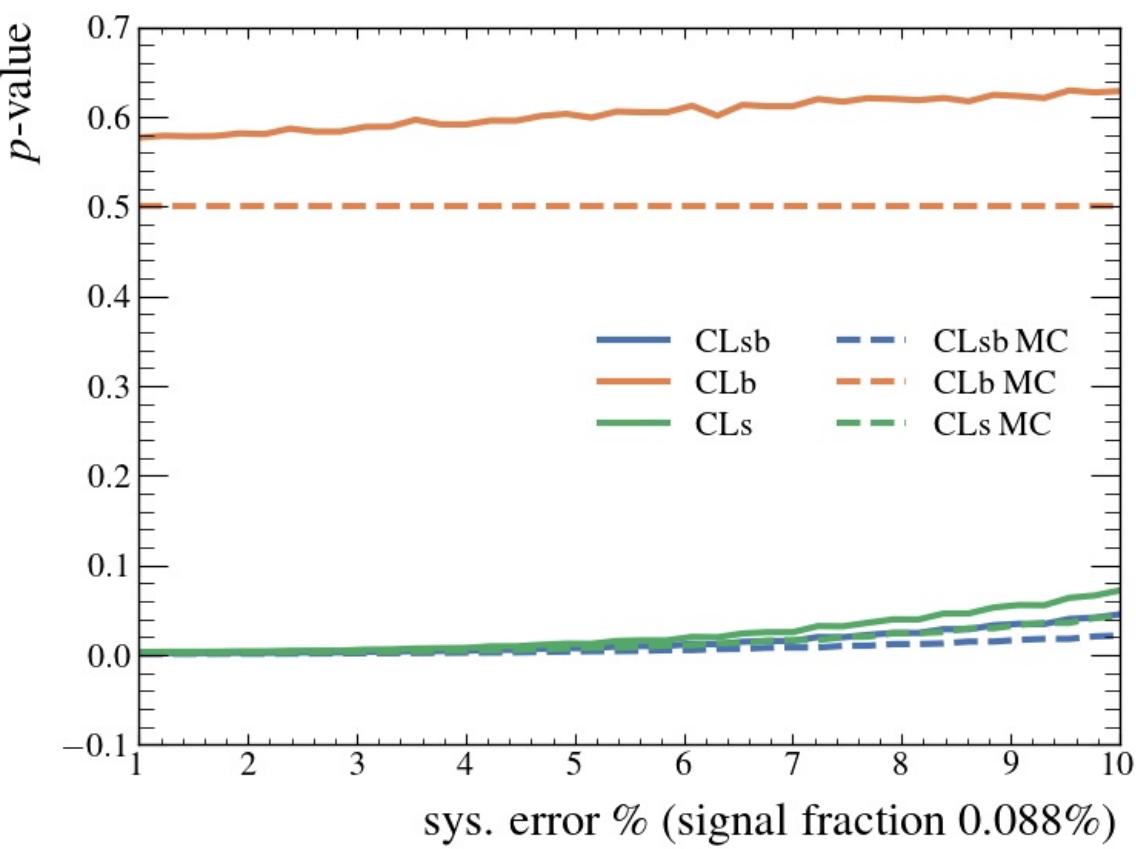


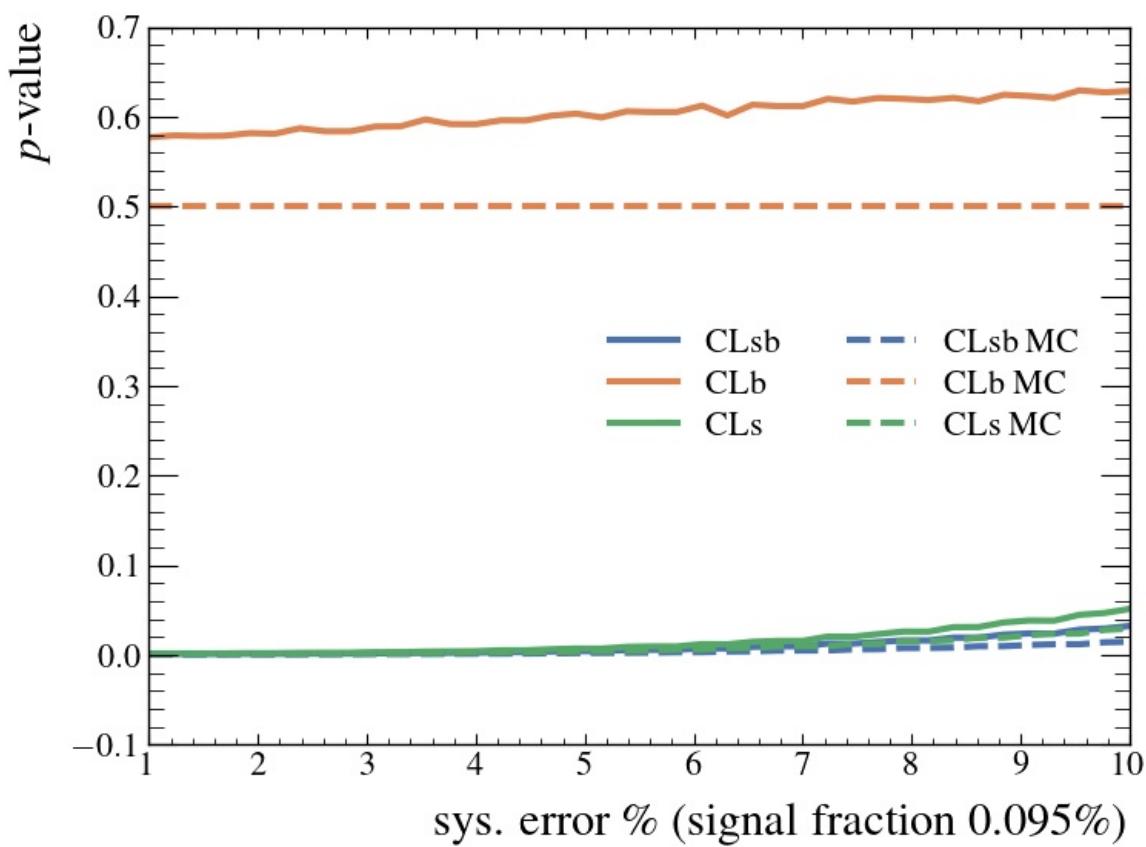
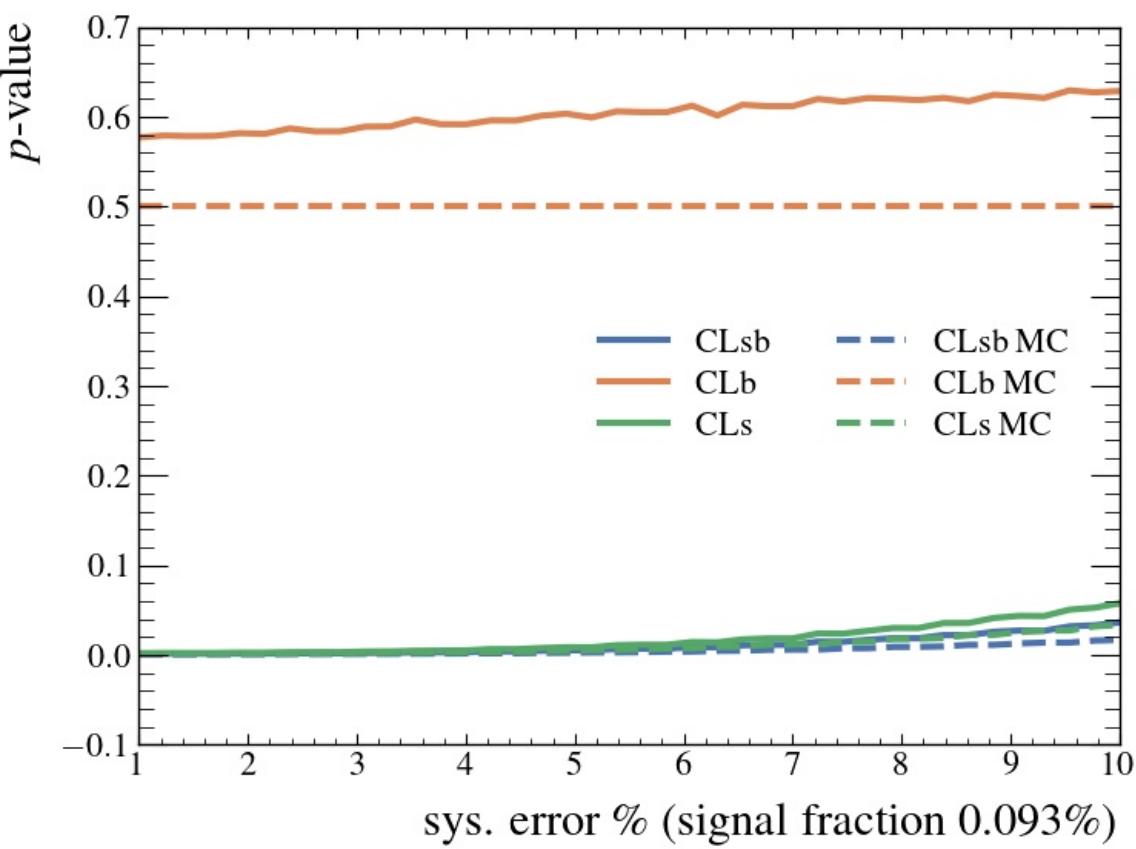


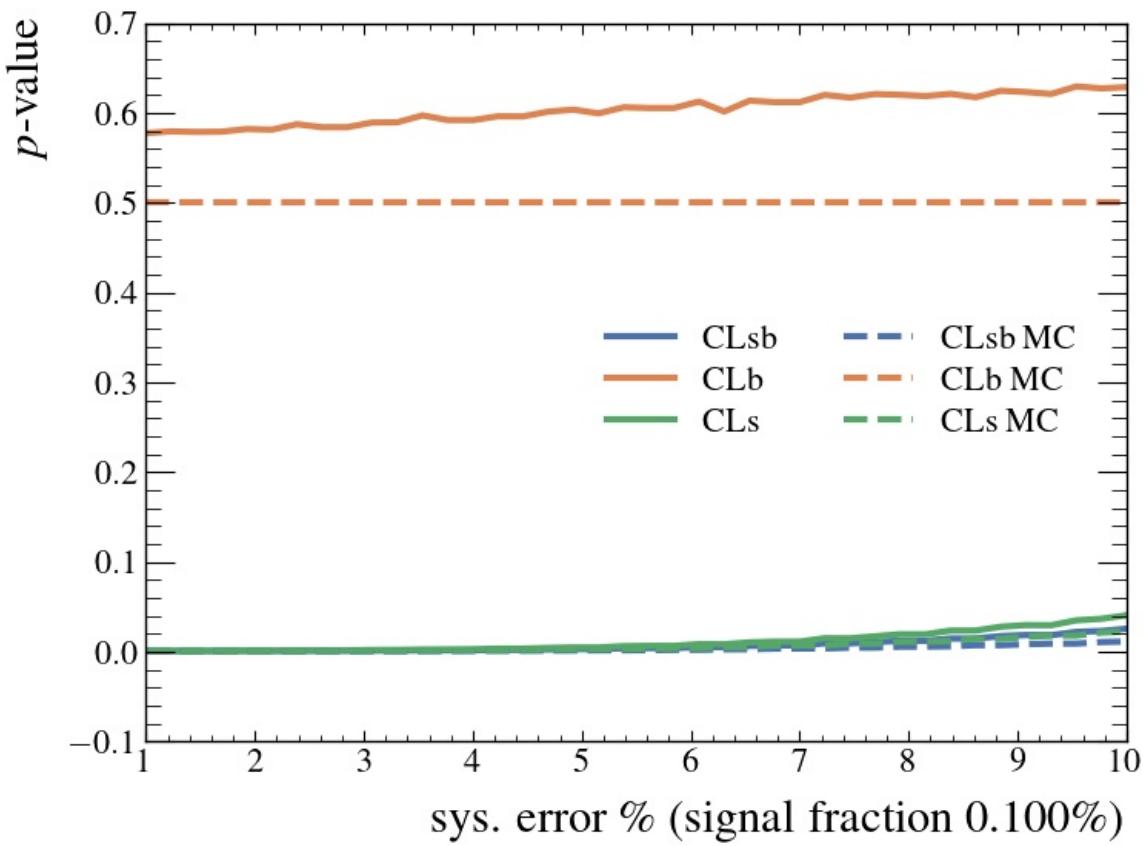
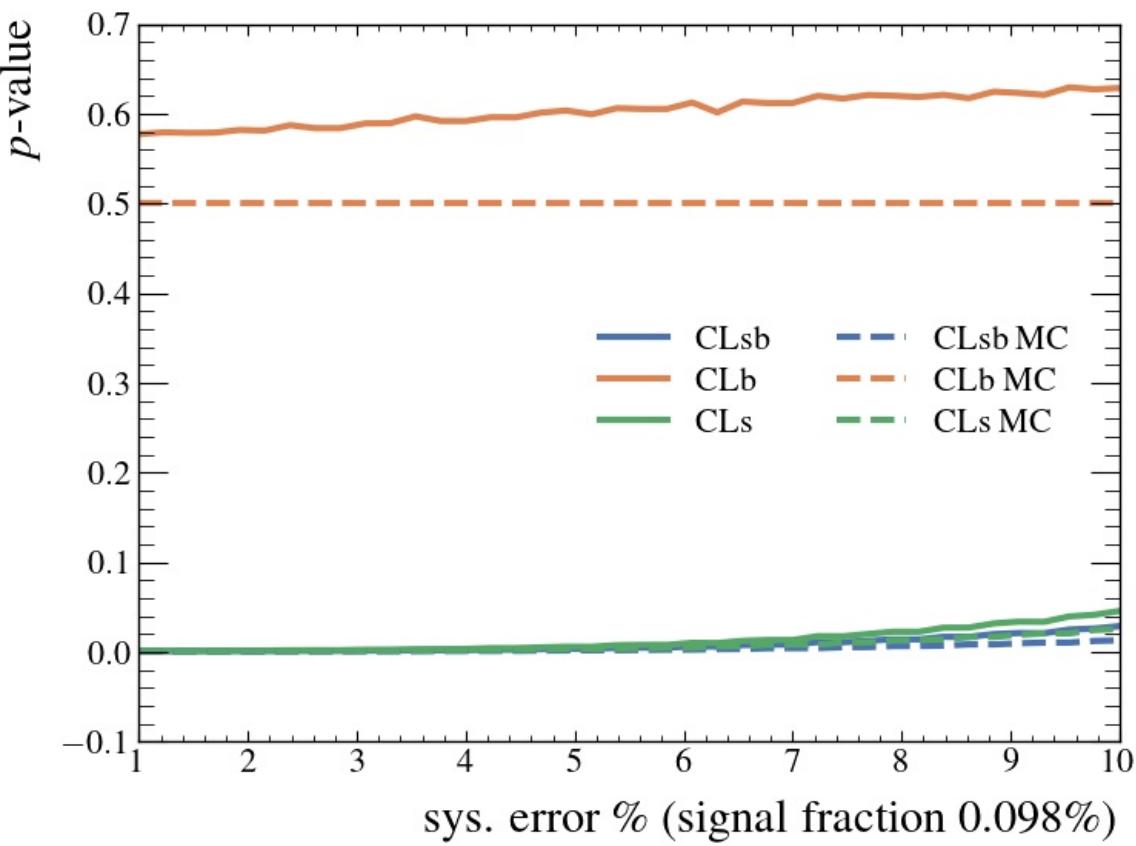












```
In [21]: # plt.plot(mc_["bkg_err"] * 100, mc_["p_sb"] - ml_["p_sb"], ls='-', c='C0', lw=3)
# plt.plot(mc_["bkg_err"] * 100, mc_["p_b"] - ml_["p_b"], ls='-', c='C1', lw=3)
# plt.plot(mc_["bkg_err"] * 100, mc_["p_s"] - ml_["p_s"], ls='-', c='C2', lw=3)

# plt.legend(["CLsb MC - CLsb", "CLb MC - CLb", "CLs MC - CLs"])

# plt.xlabel("sys. error \% (signal fraction {:.2f}\%)".format(f), fontsize=22)
# plt.ylabel("$p$-value difference", fontsize=22)
# plt.tight_layout()

# plt.savefig(saved + "CLs_q0_mu0_bkg_errs_diff.pdf")
```

Constant background error

```

In [22]: for i, f in enumerate(bkg_errs):
    ml_ = ml_res[ml_res["bkg_err"] == f]
    mc_ = mc_res[mc_res["bkg_err"] == f]

    plt.plot(ml_["sig_frac"] * 100, ml_["p_sb"], c="C0", lw=3)
    plt.plot(ml_["sig_frac"] * 100, ml_["p_b"], c="C1", lw=3)
    plt.plot(ml_["sig_frac"] * 100, ml_["p_s"], c="C2", lw=3)

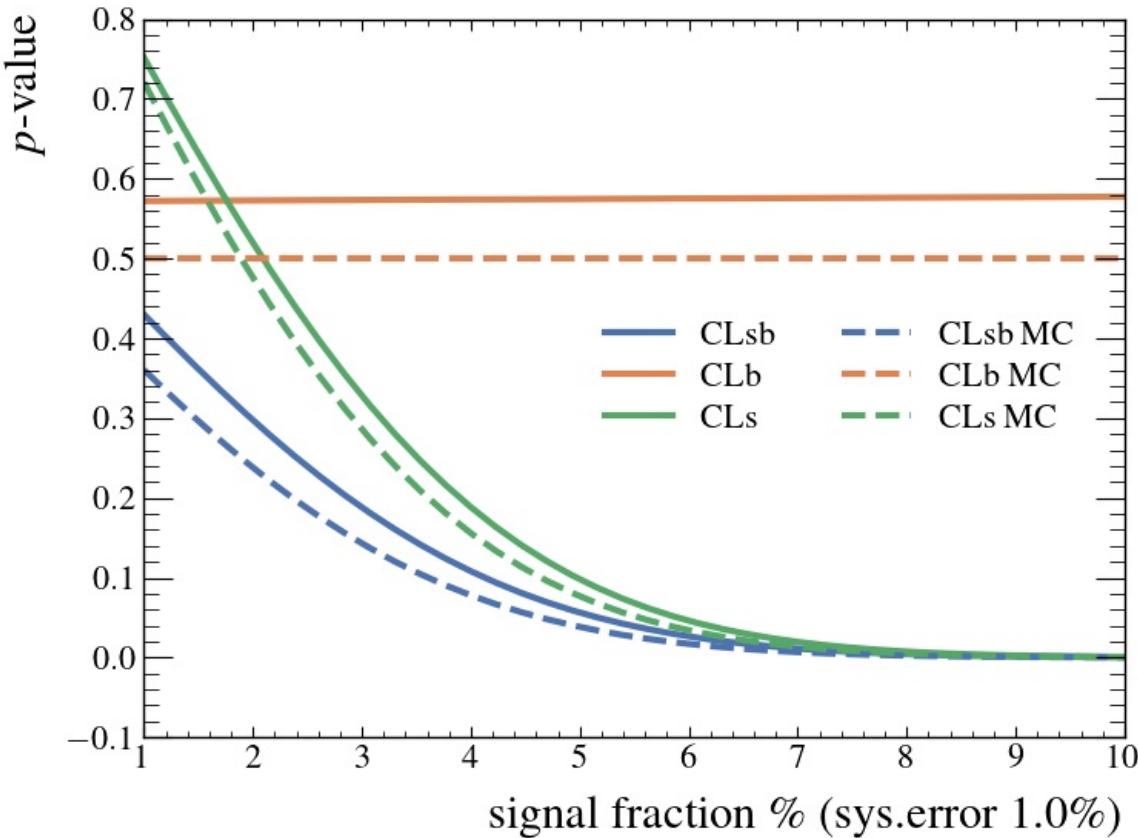
    plt.plot(mc_["sig_frac"] * 100, mc_["p_sb"], ls='--', c='C0', lw=3)
    plt.plot(mc_["sig_frac"] * 100, mc_["p_b"], ls='--', c='C1', lw=3)
    plt.plot(mc_["sig_frac"] * 100, mc_["p_s"], ls='--', c='C2', lw=3)

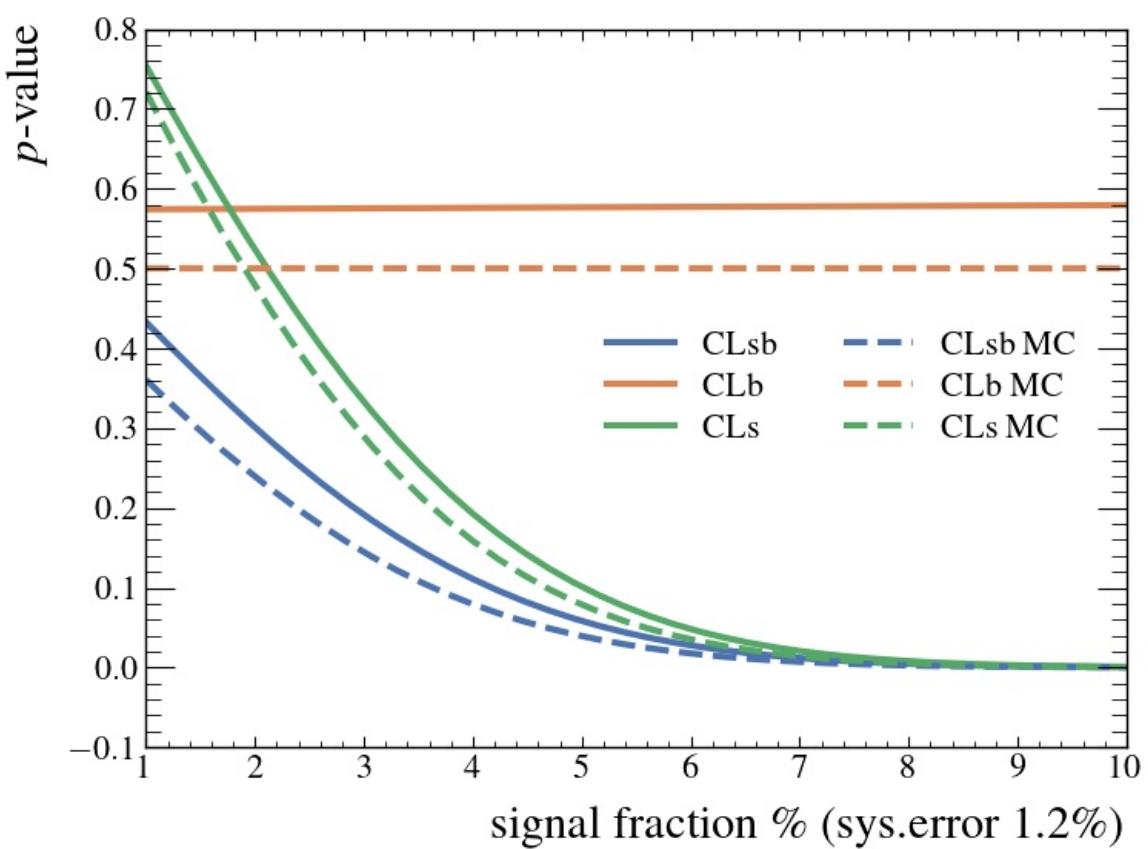
plt.legend(["CLsb", "CLb", "CLs", "CLsb MC", "CLb MC", "CLs MC"], ncol=2)

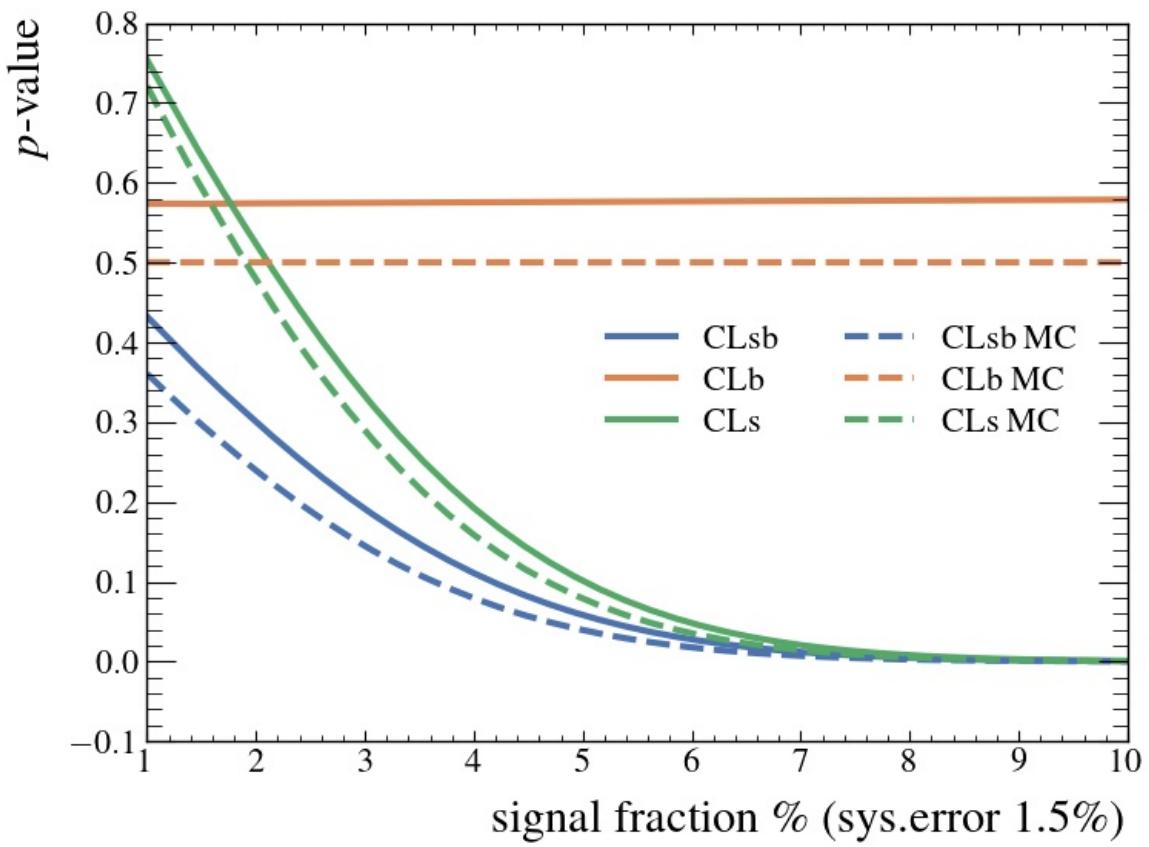
plt.xlabel(f"signal fraction \% (sys.error {100*f:.1f}\%)", fontsize=22)
plt.ylabel("$p$-value", fontsize=22)
plt.tight_layout()

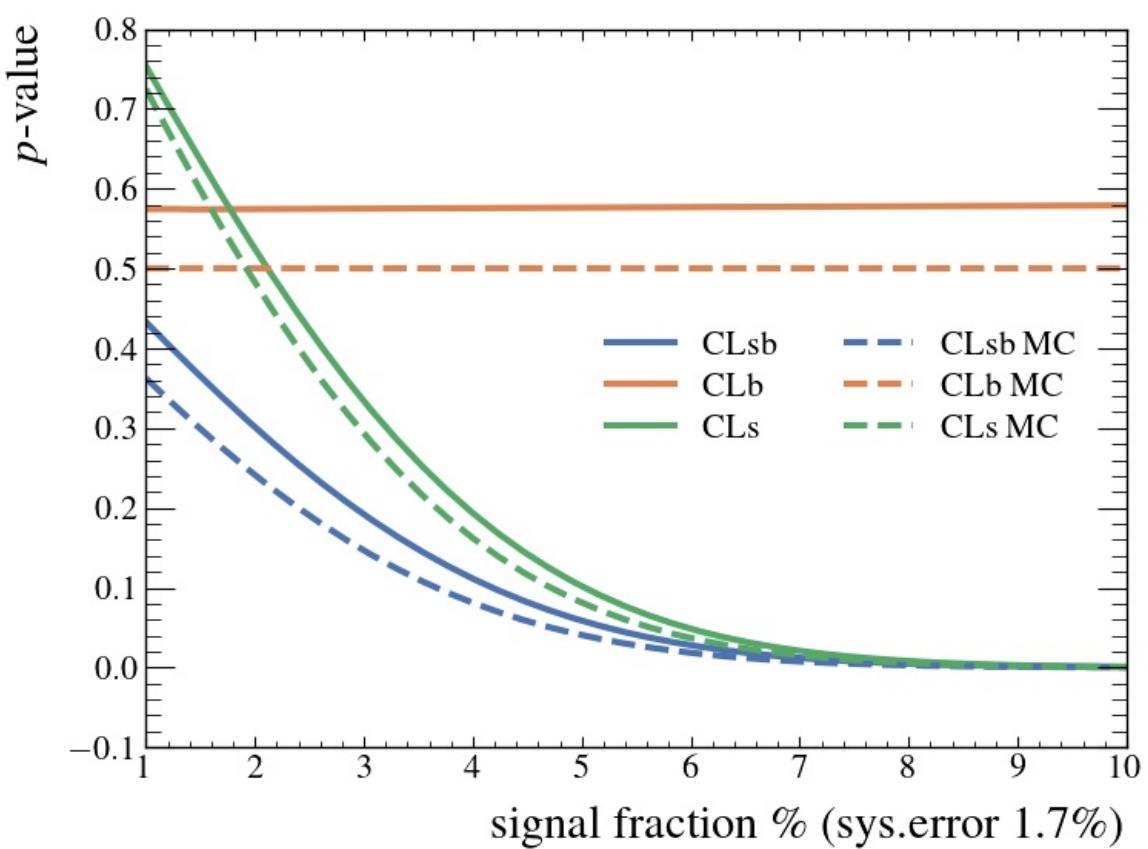
plt.savefig(saved + f"CLs_q0_mu0_sig_fracs_{i}.pdf")
plt.show()

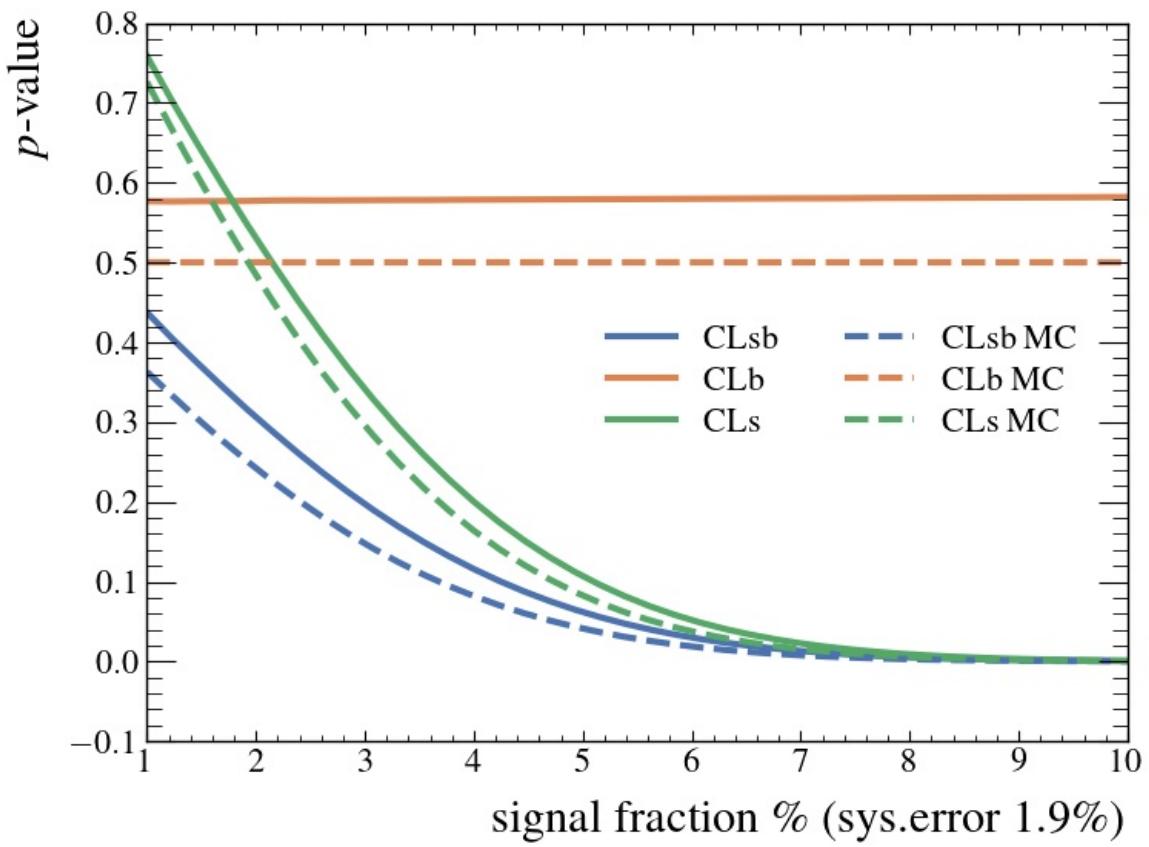
```

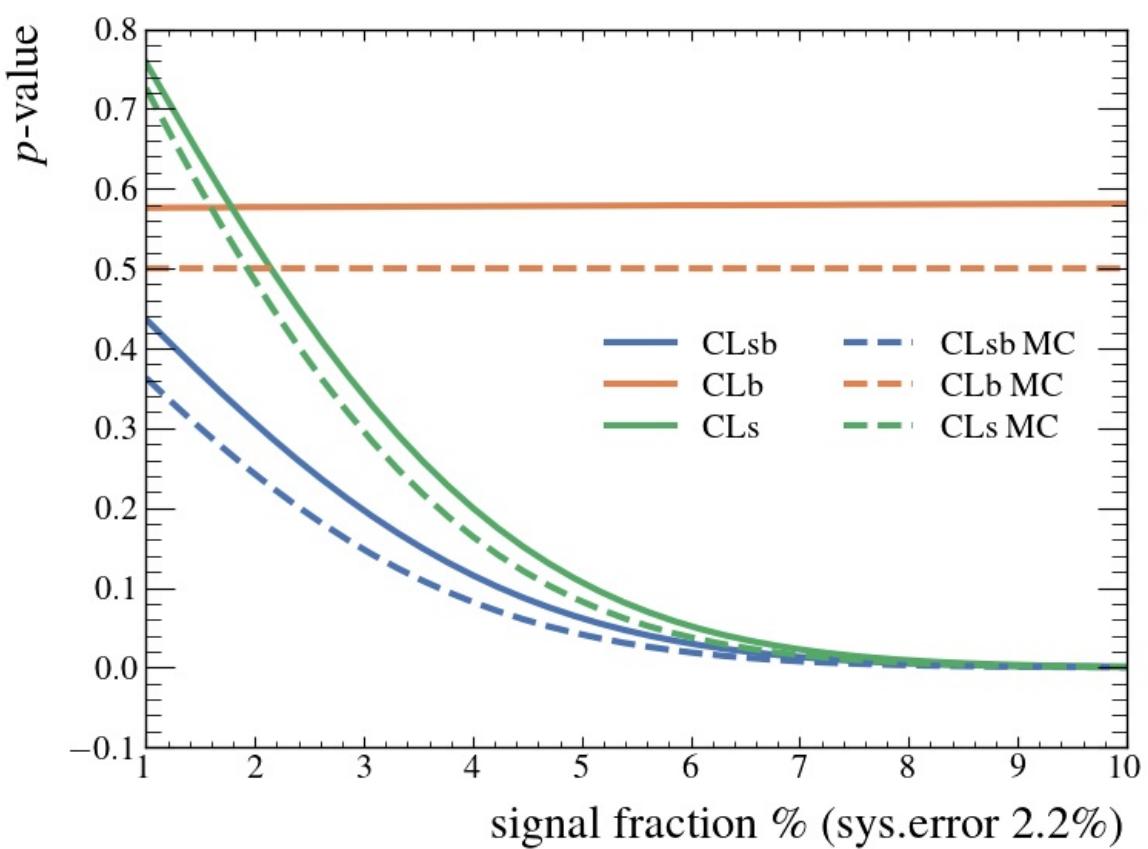


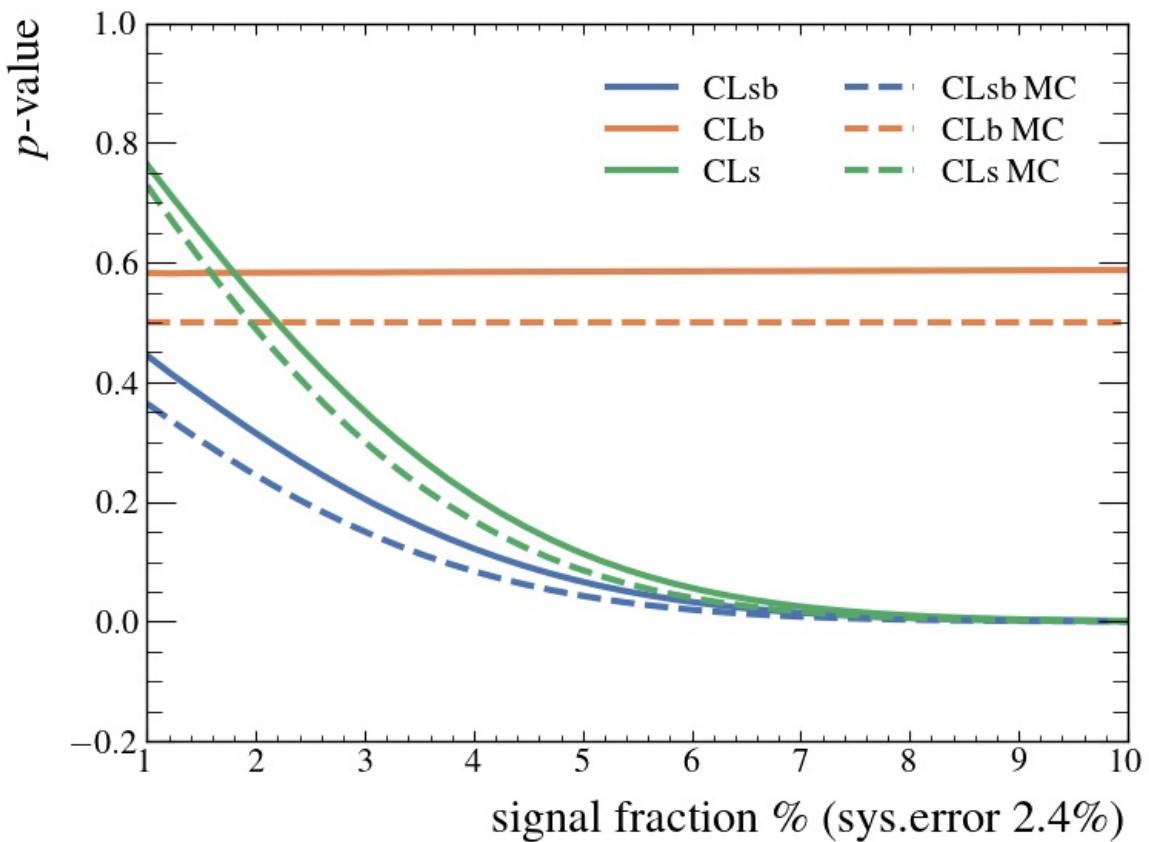


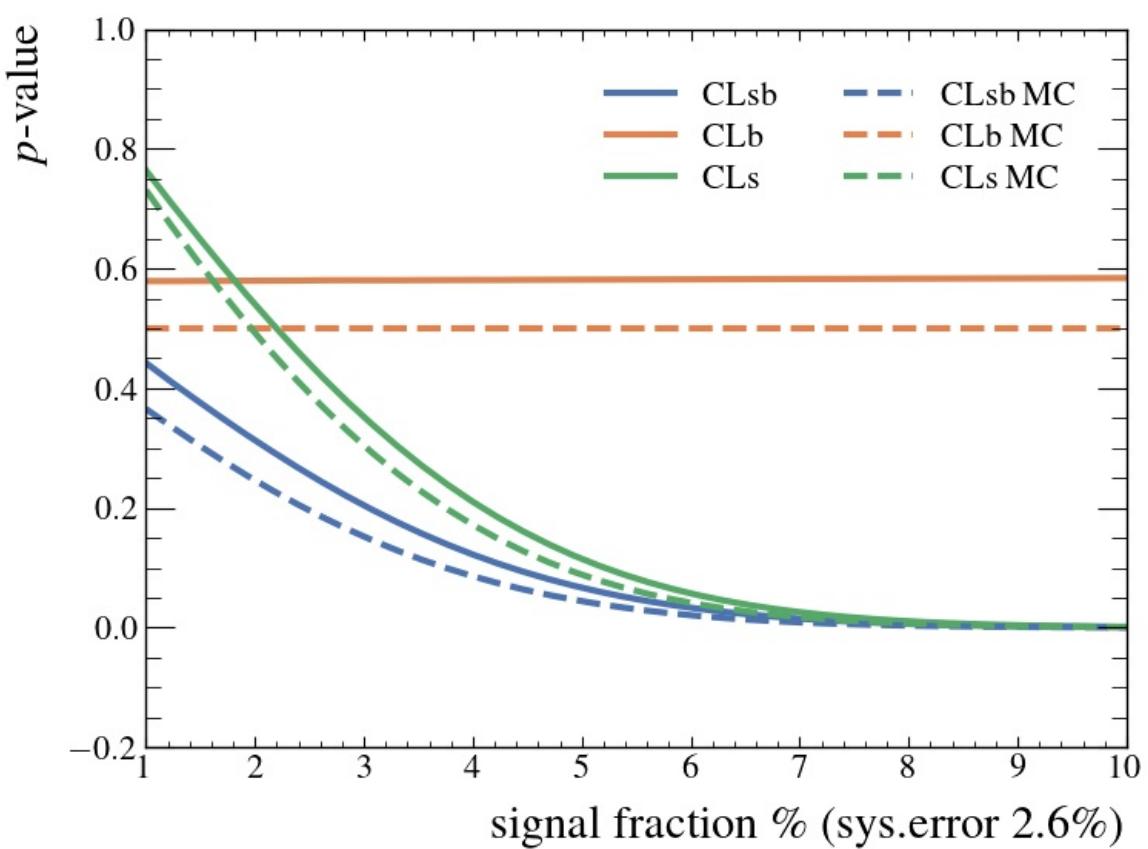


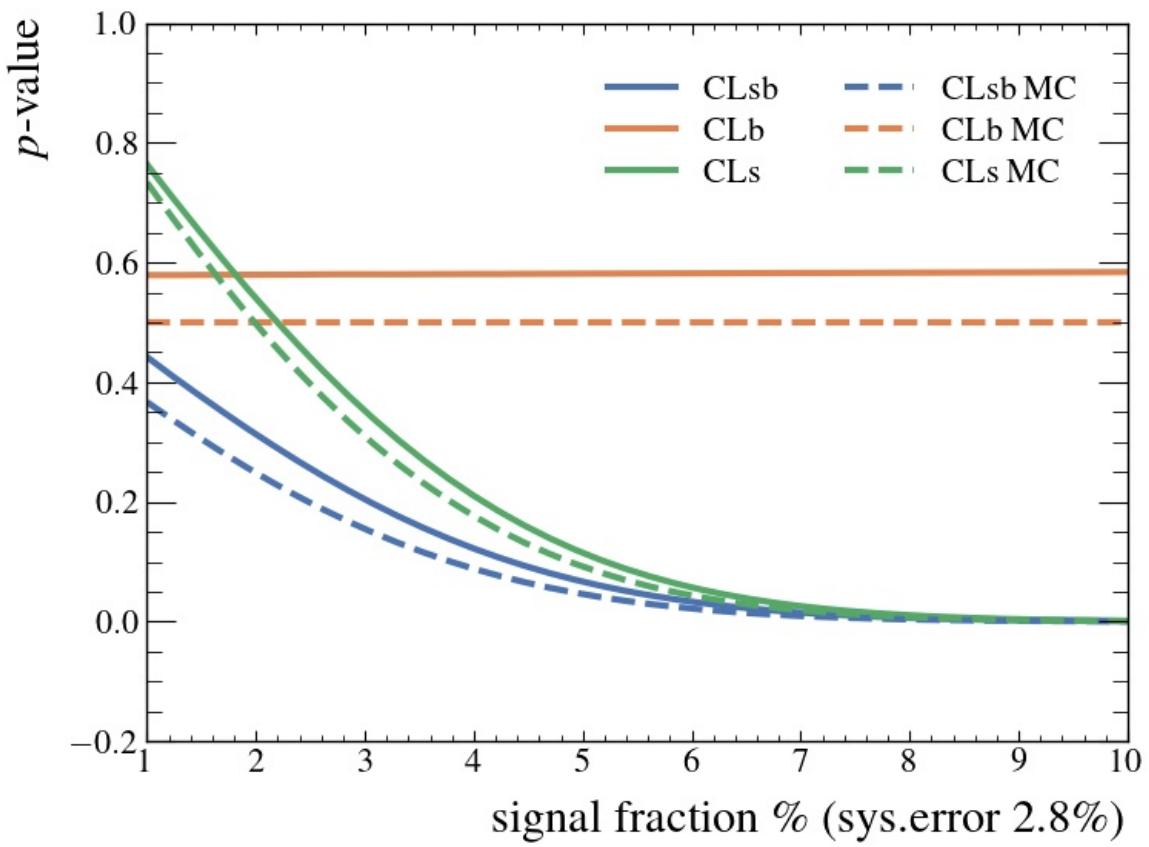


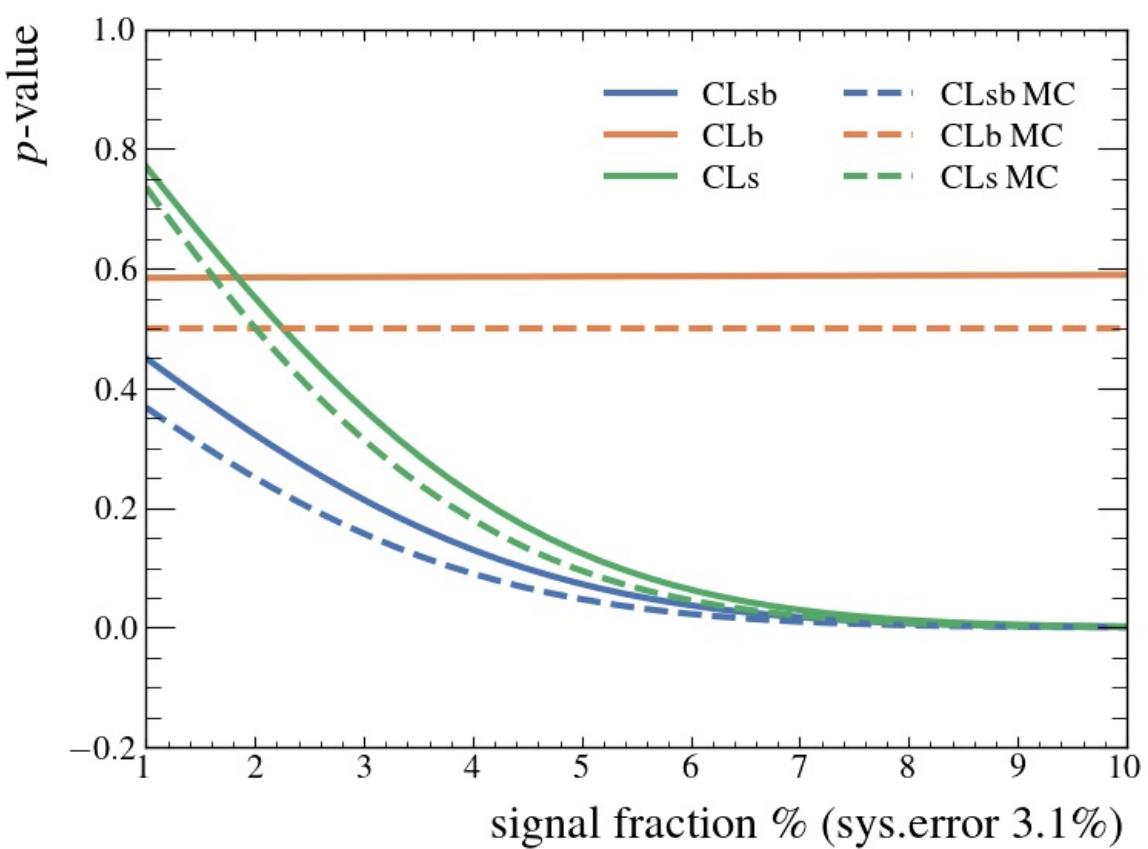


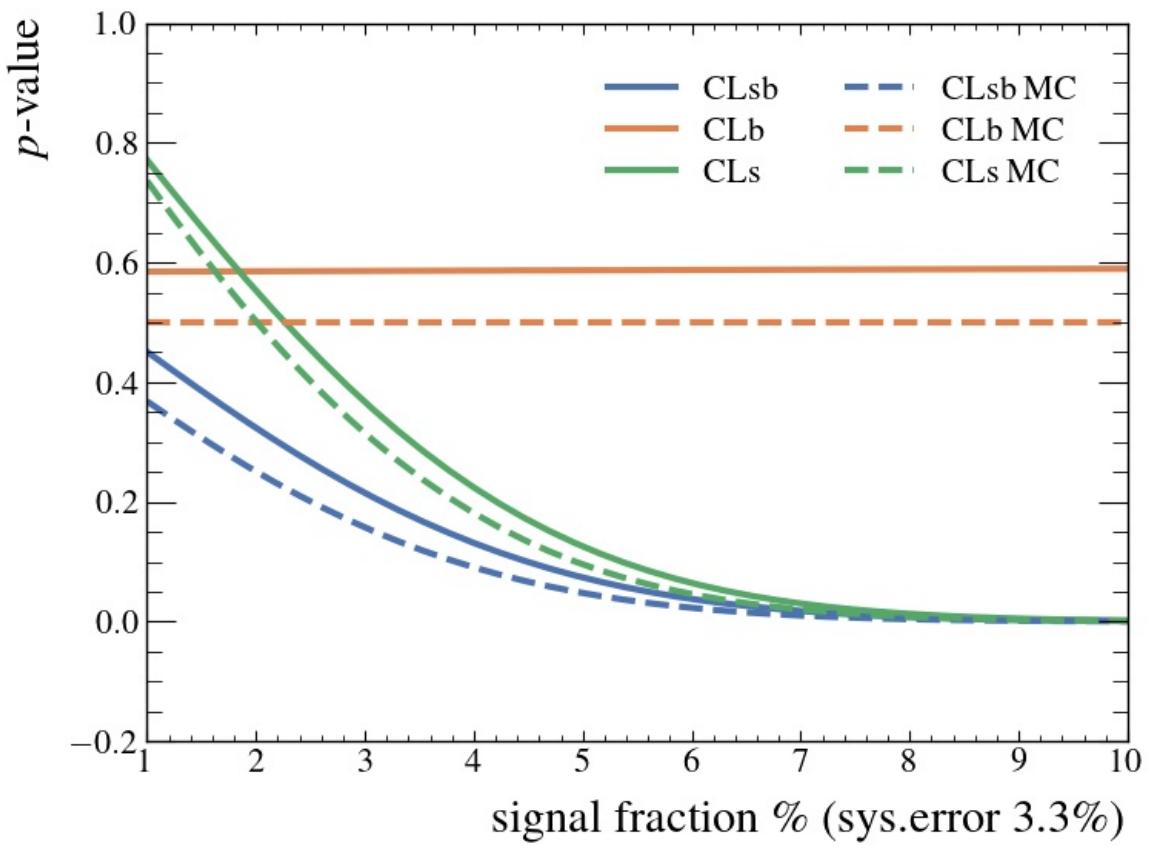


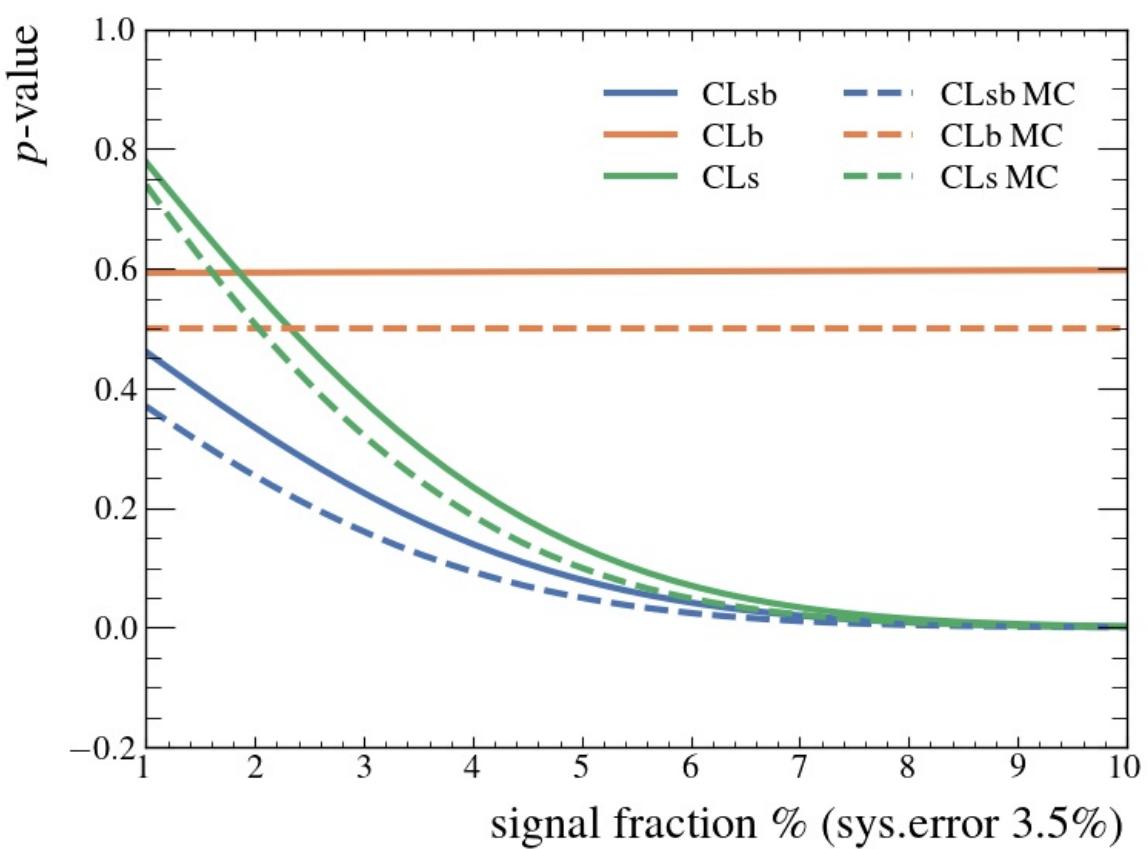


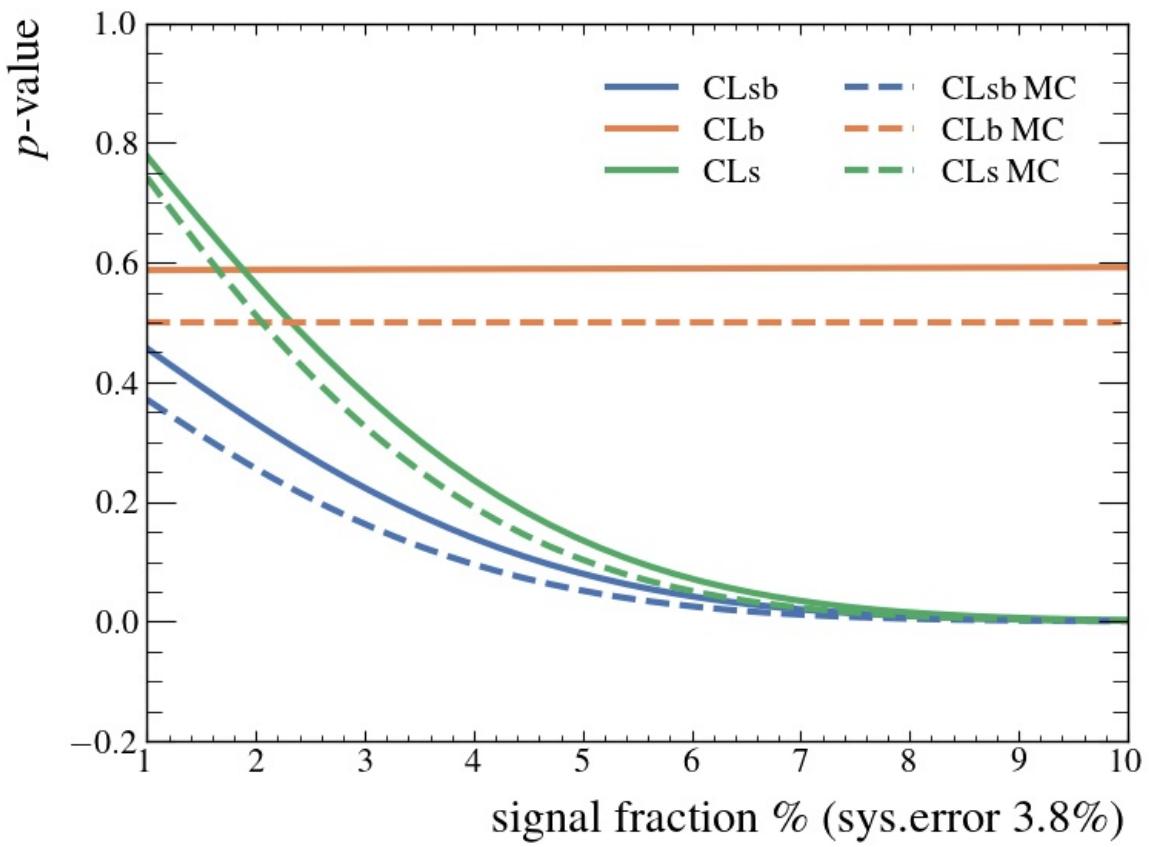


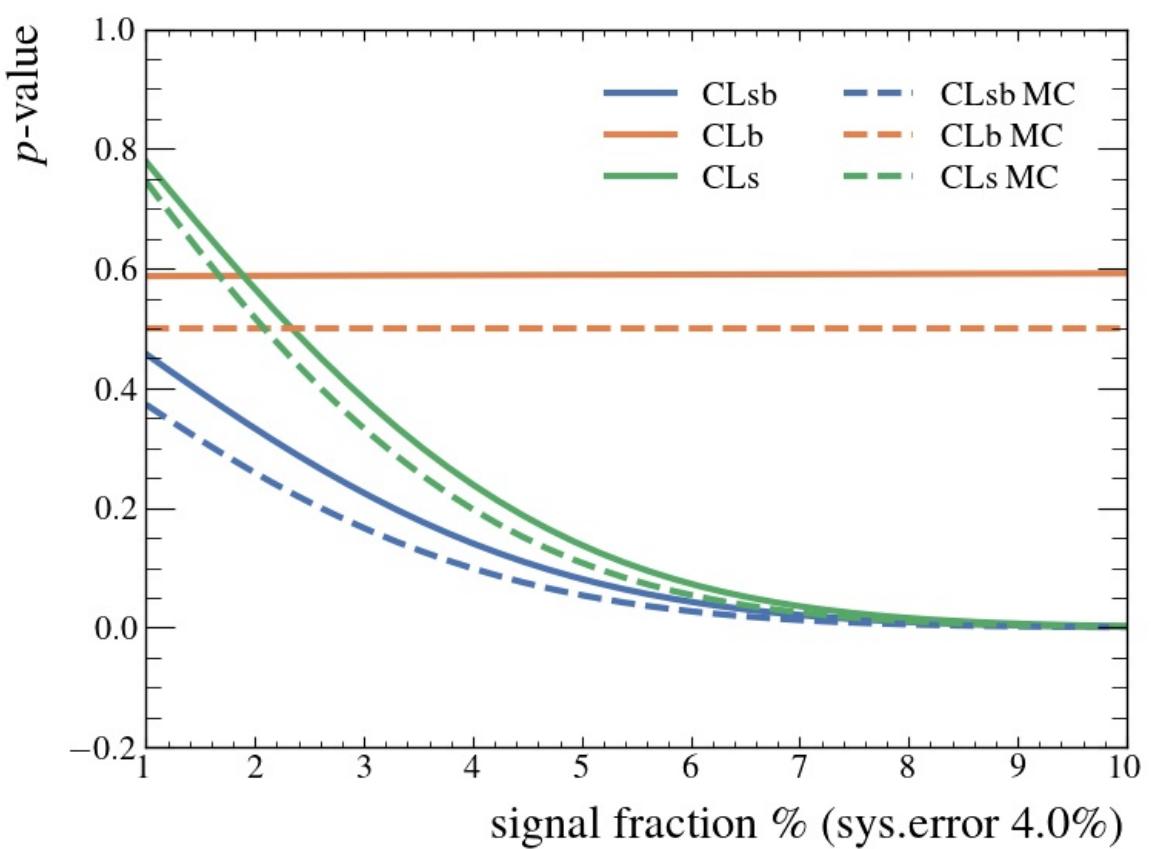


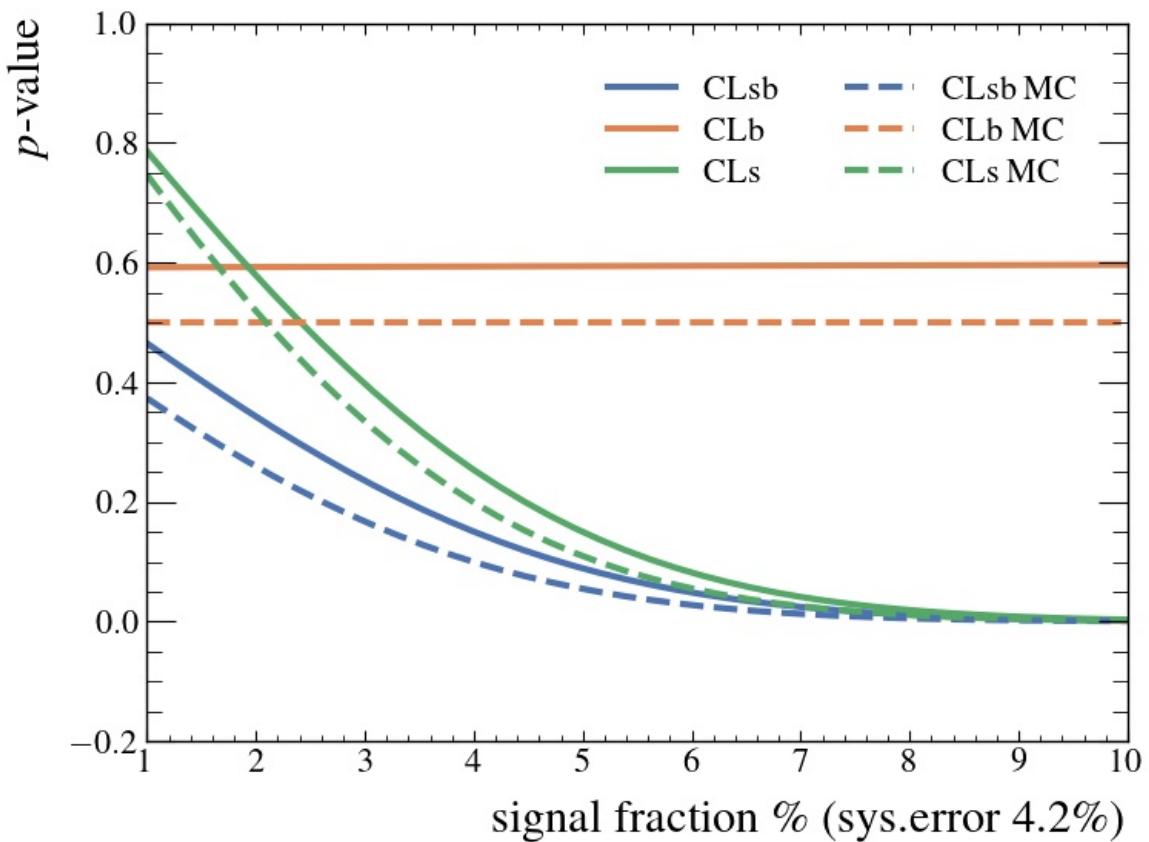


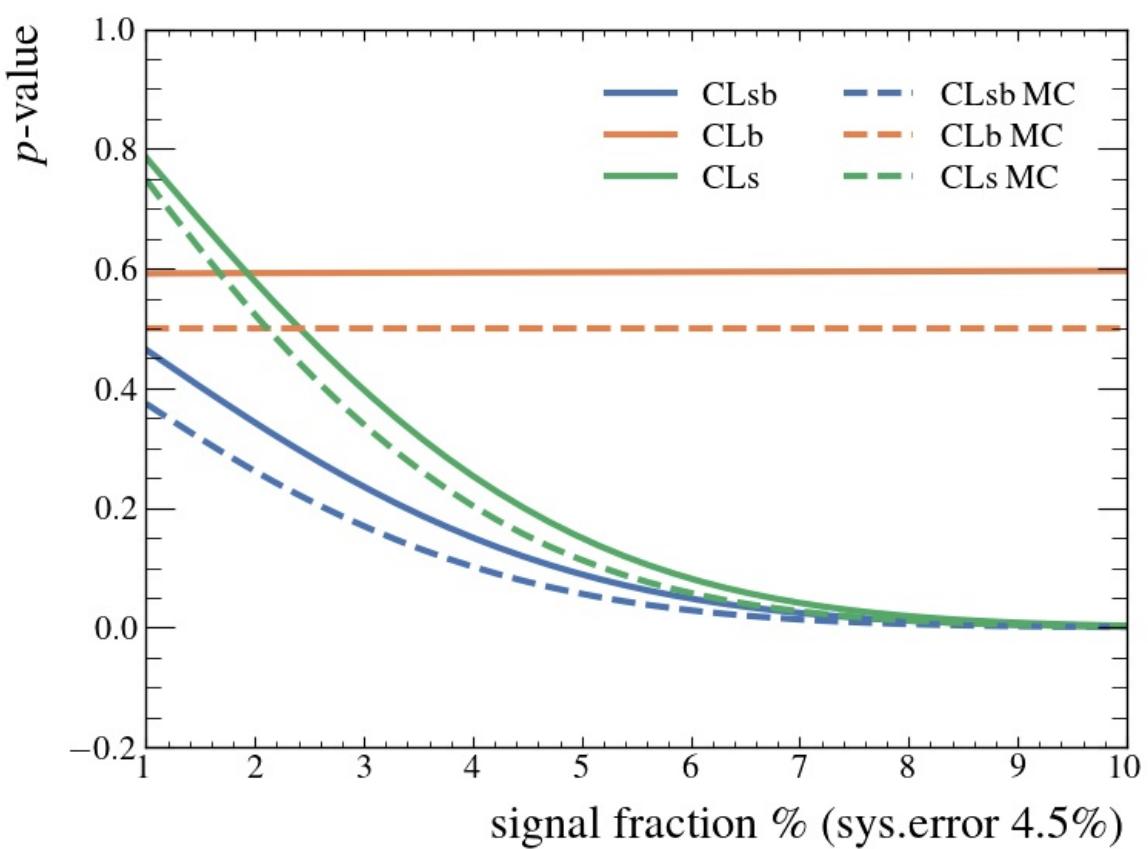


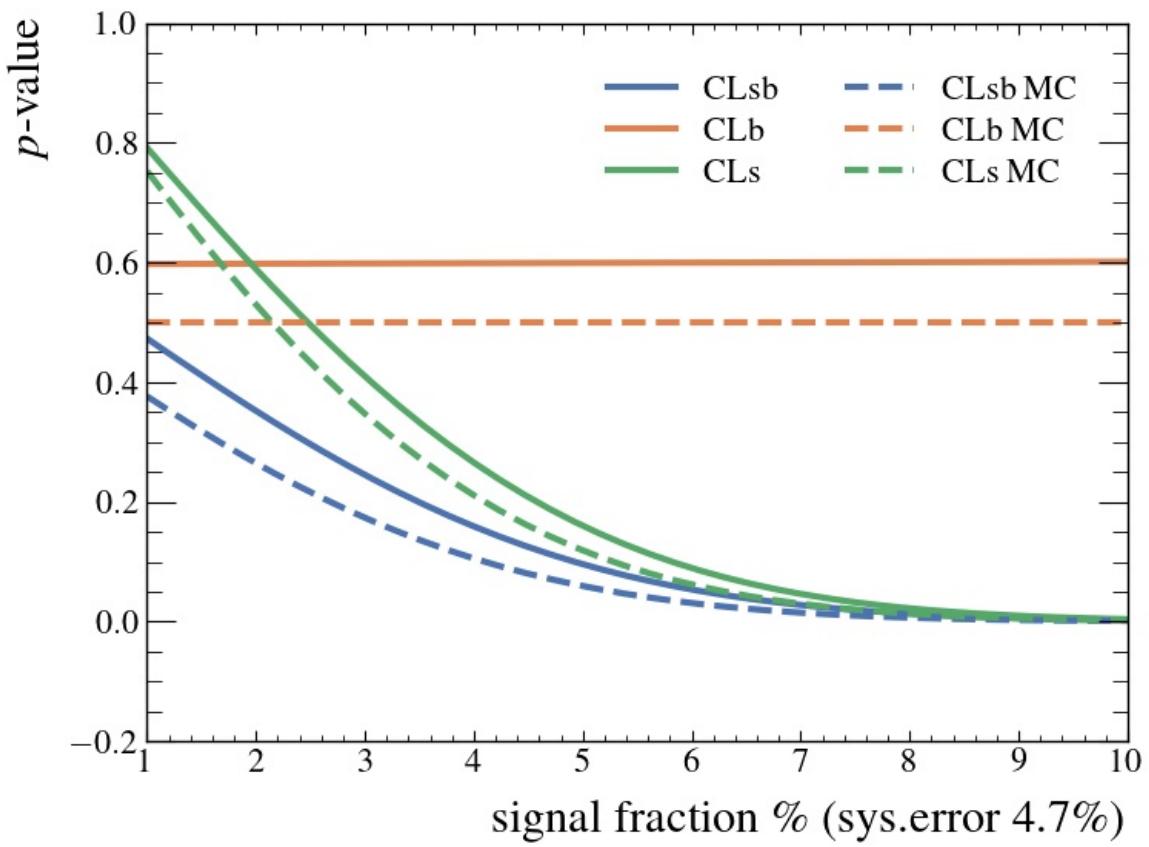


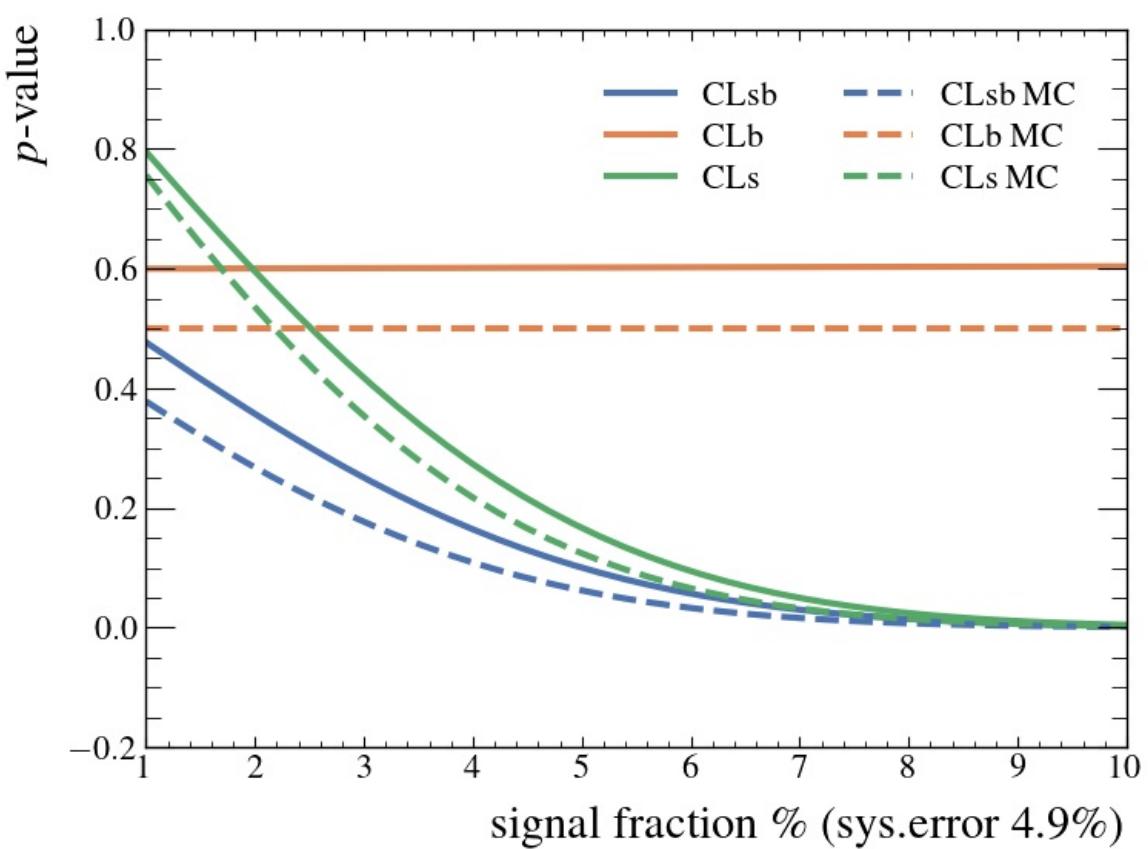


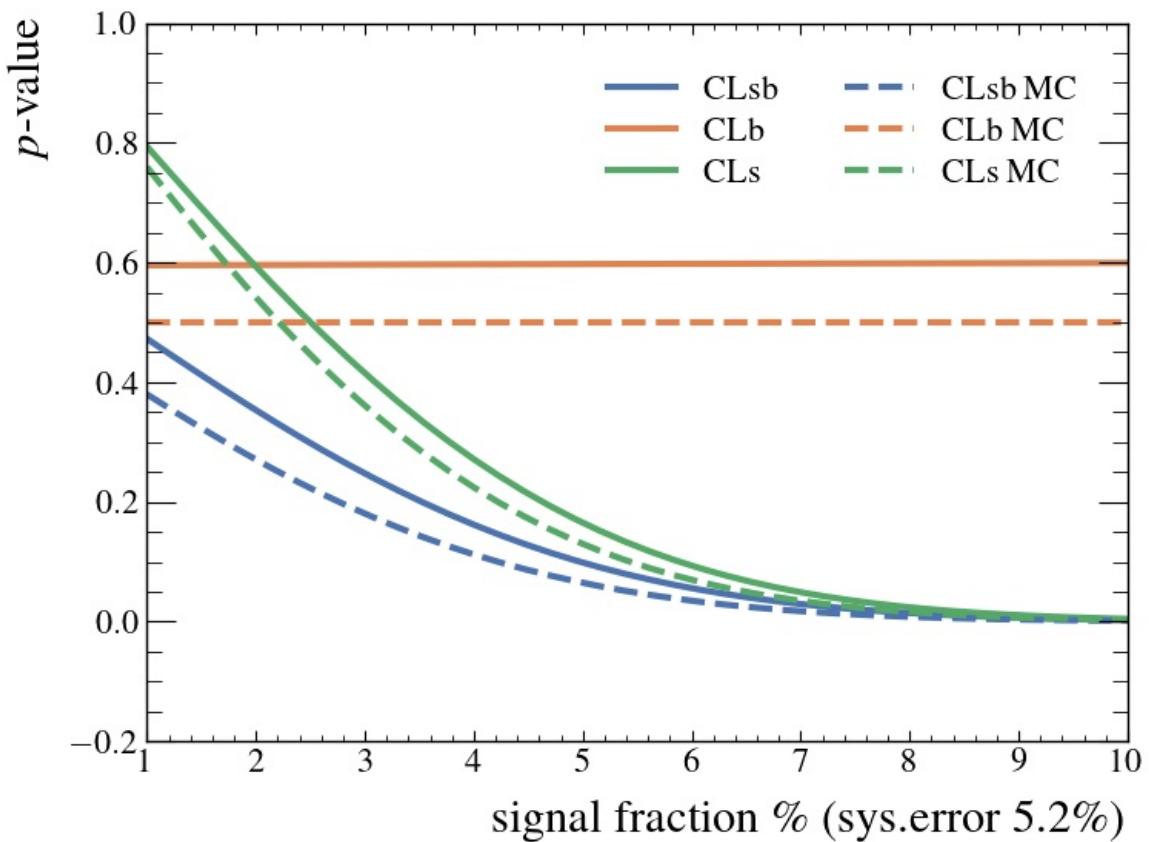


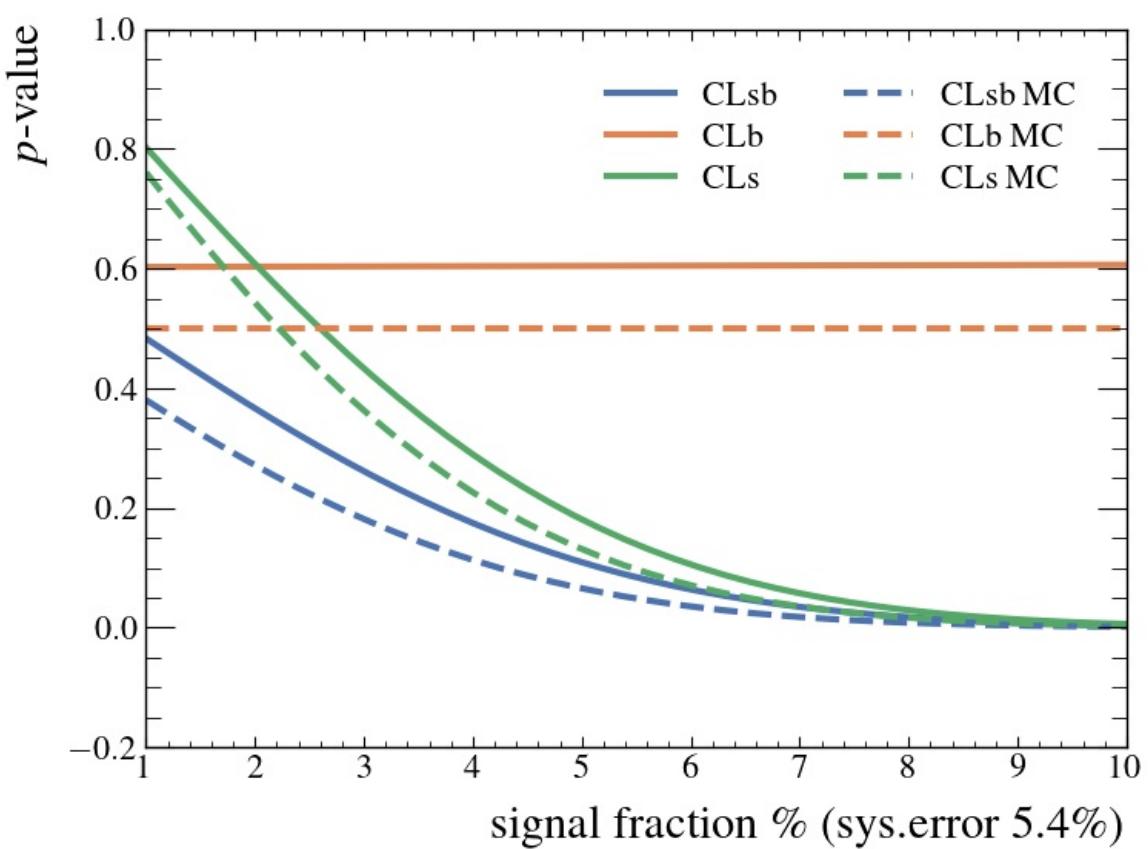


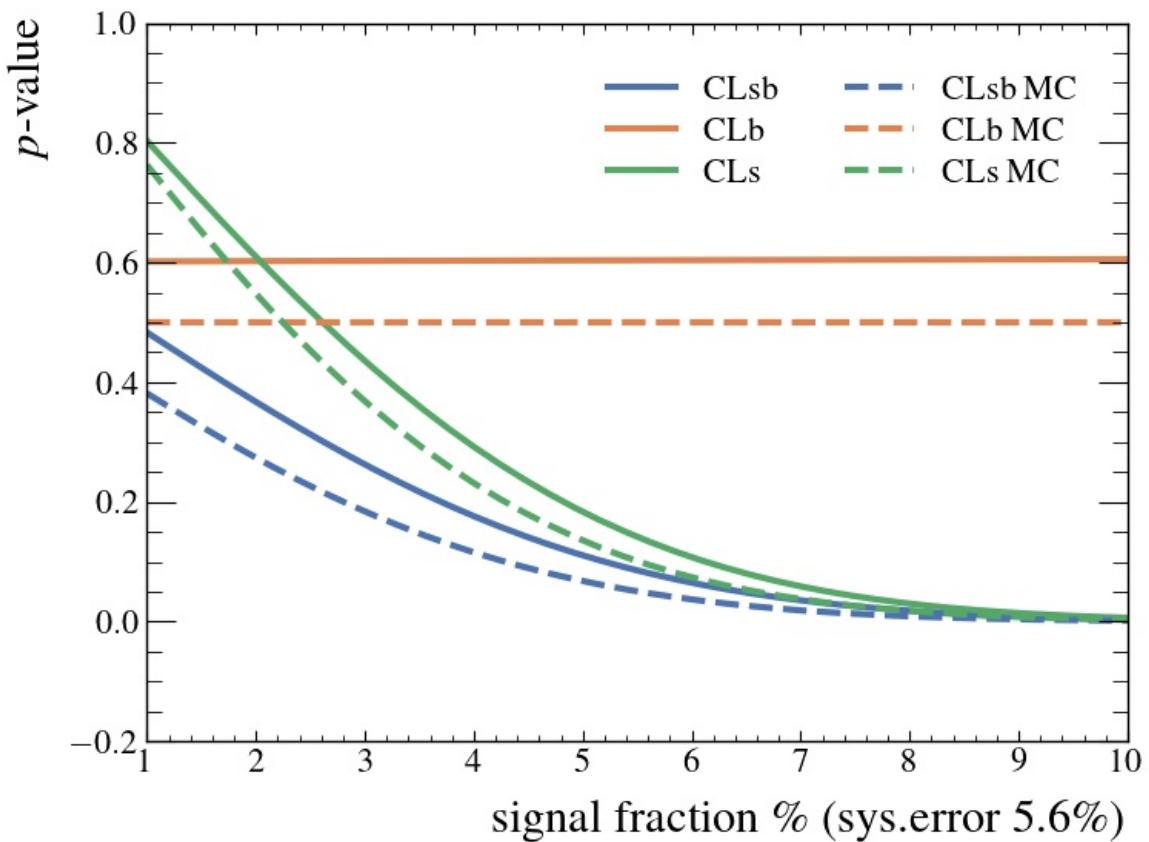


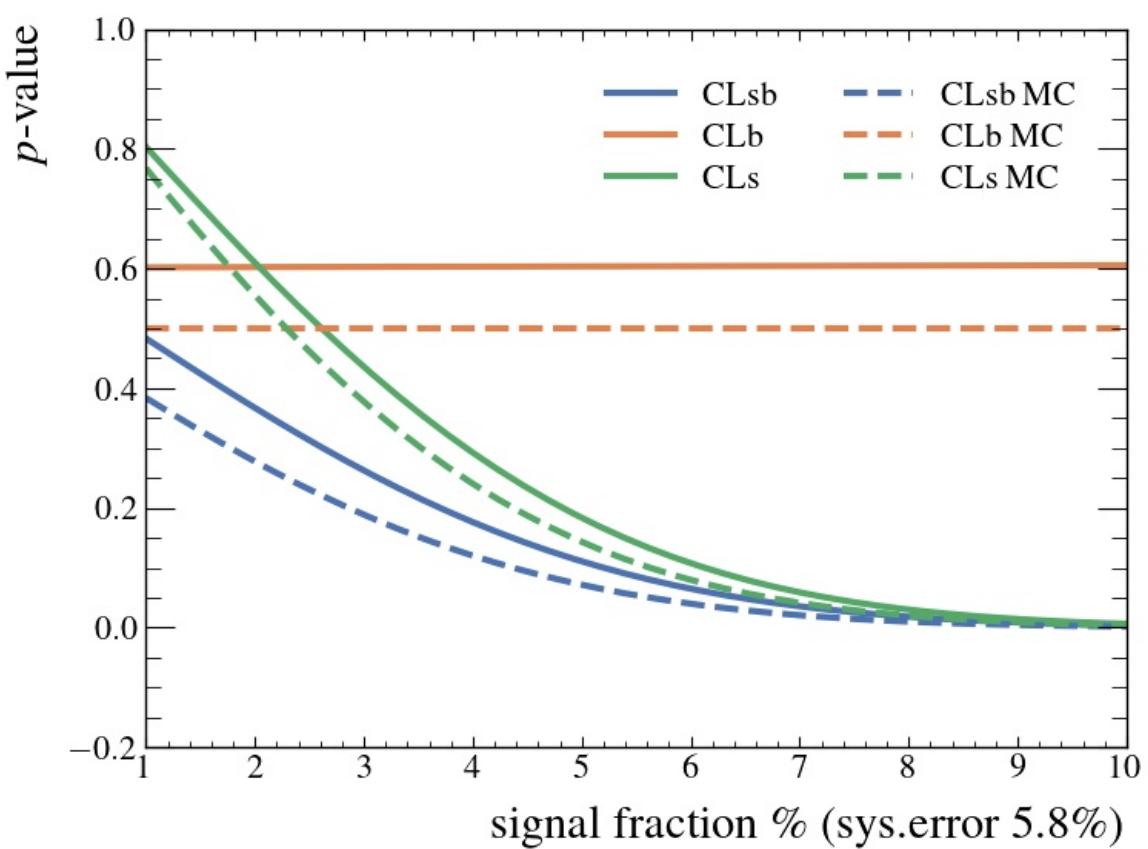


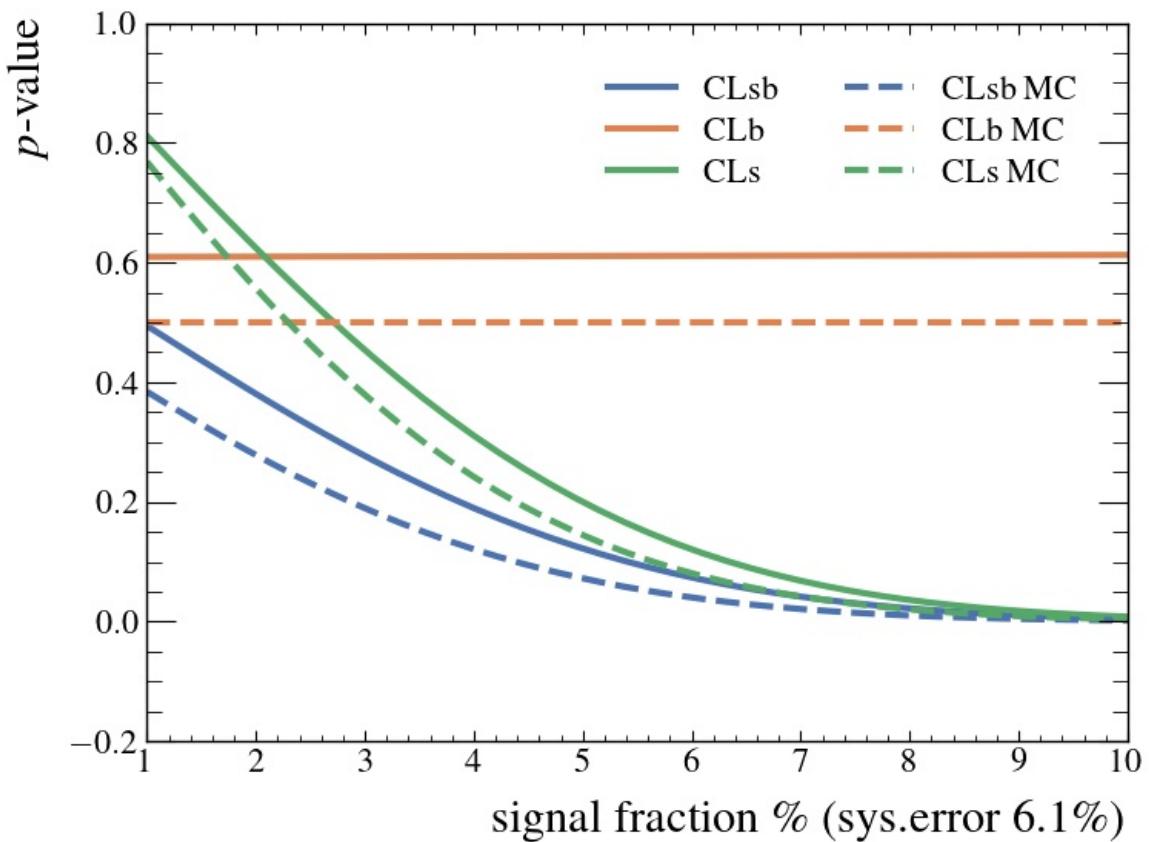


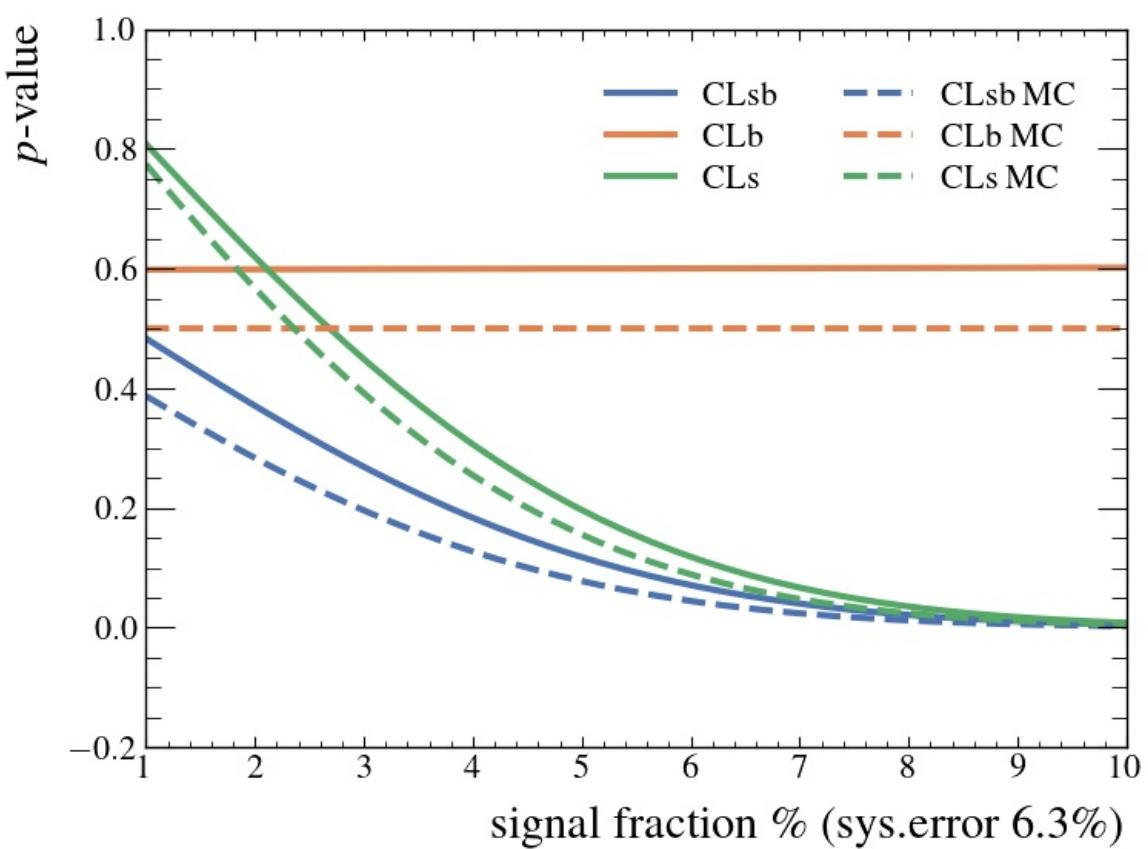


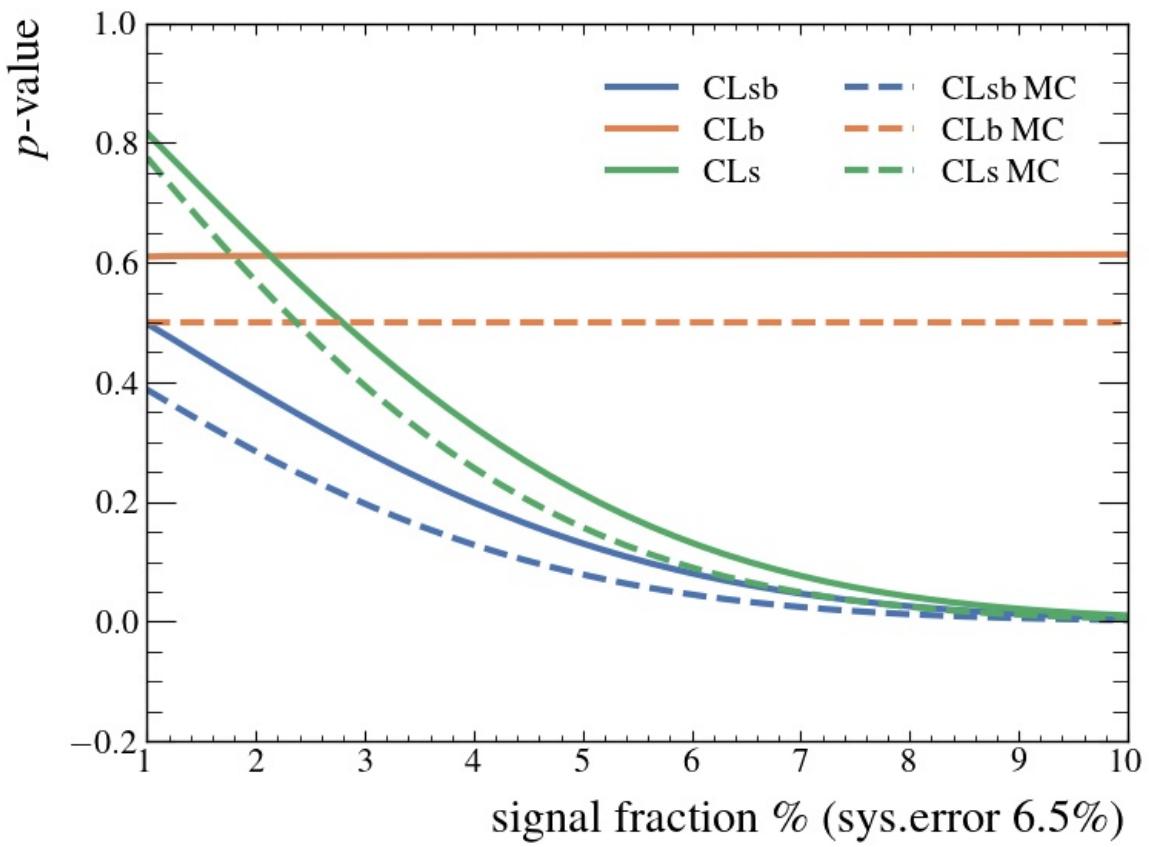


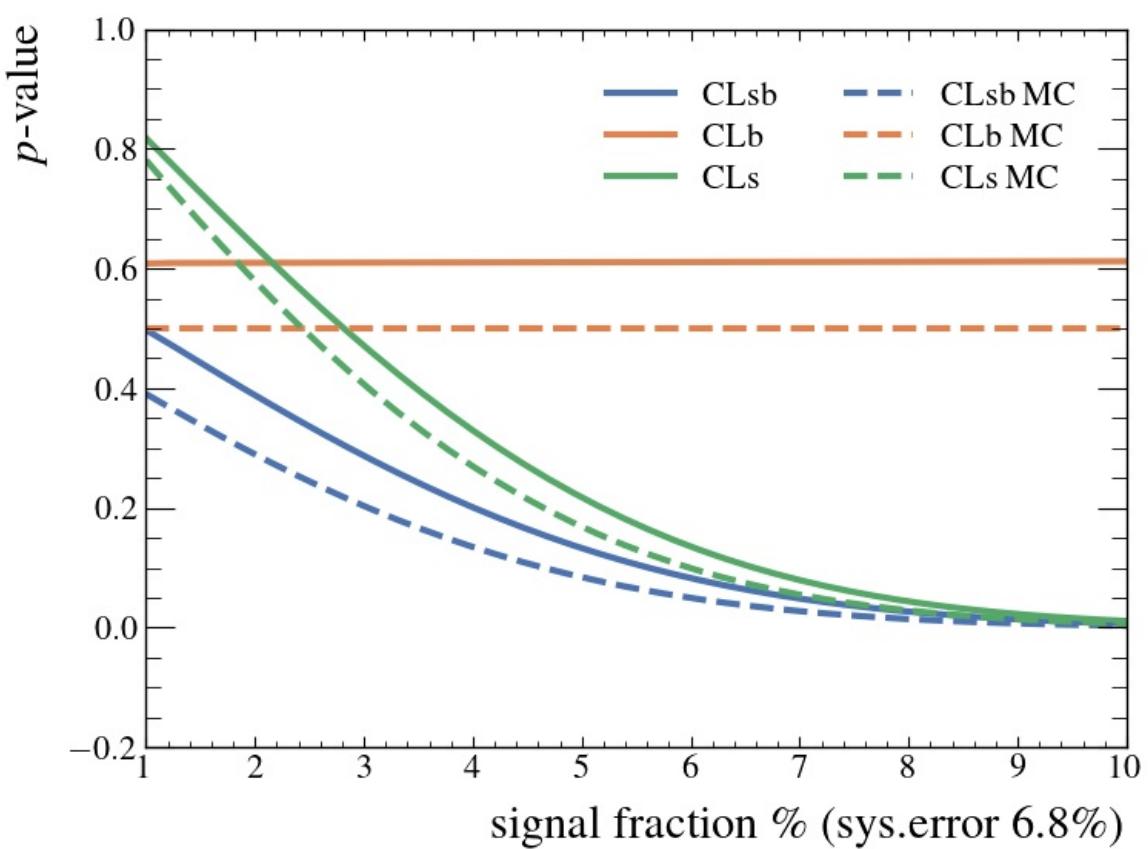


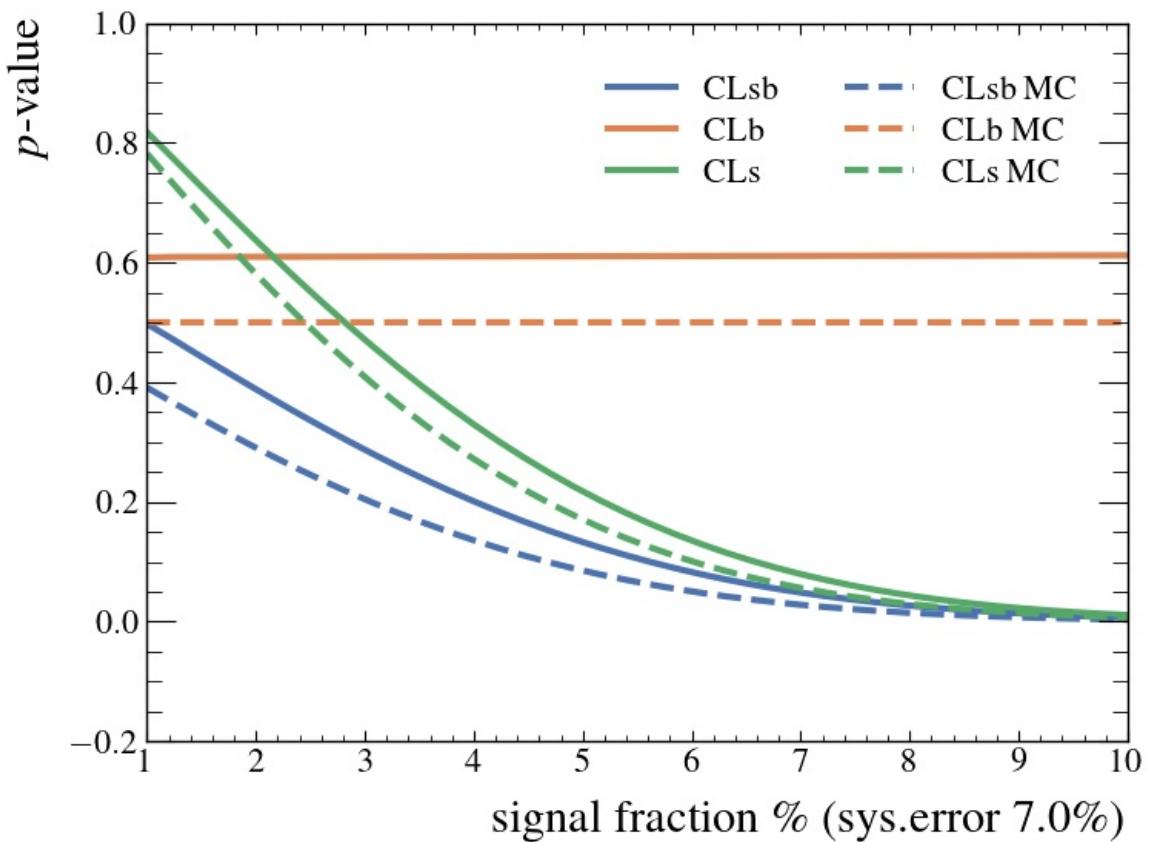


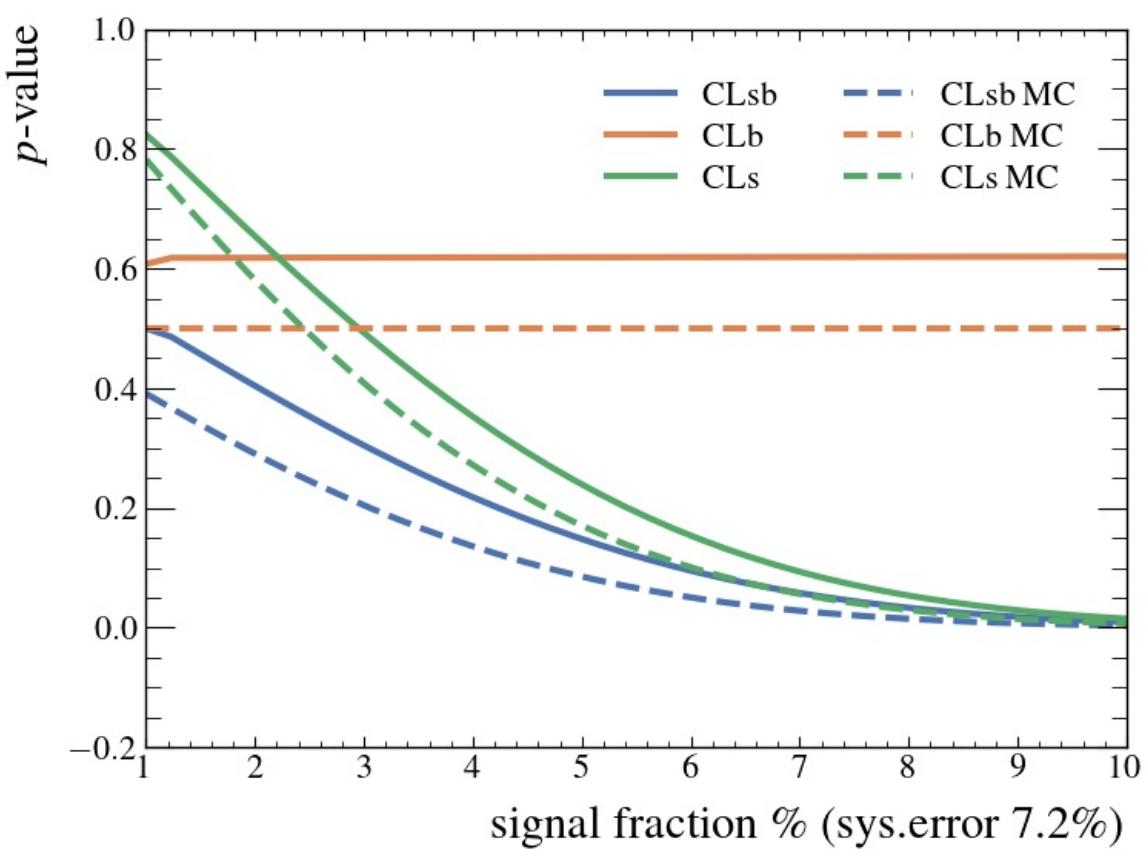


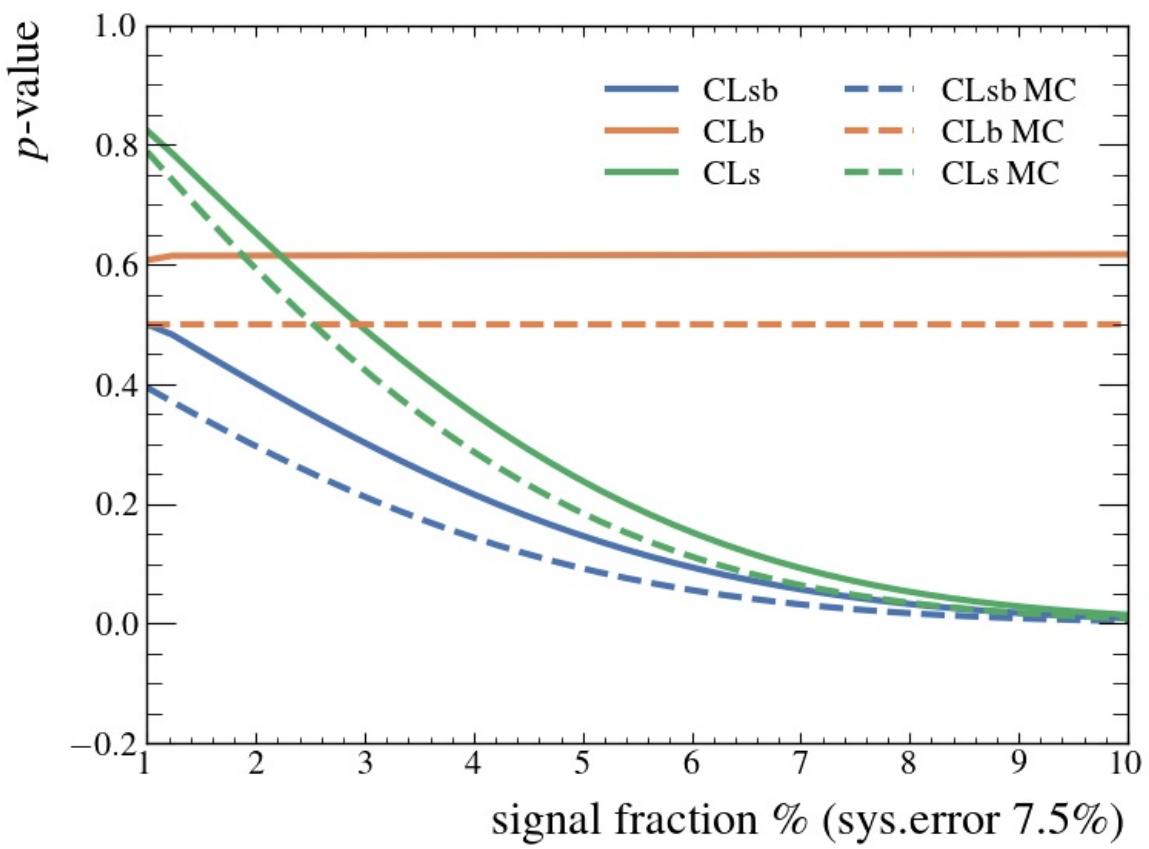


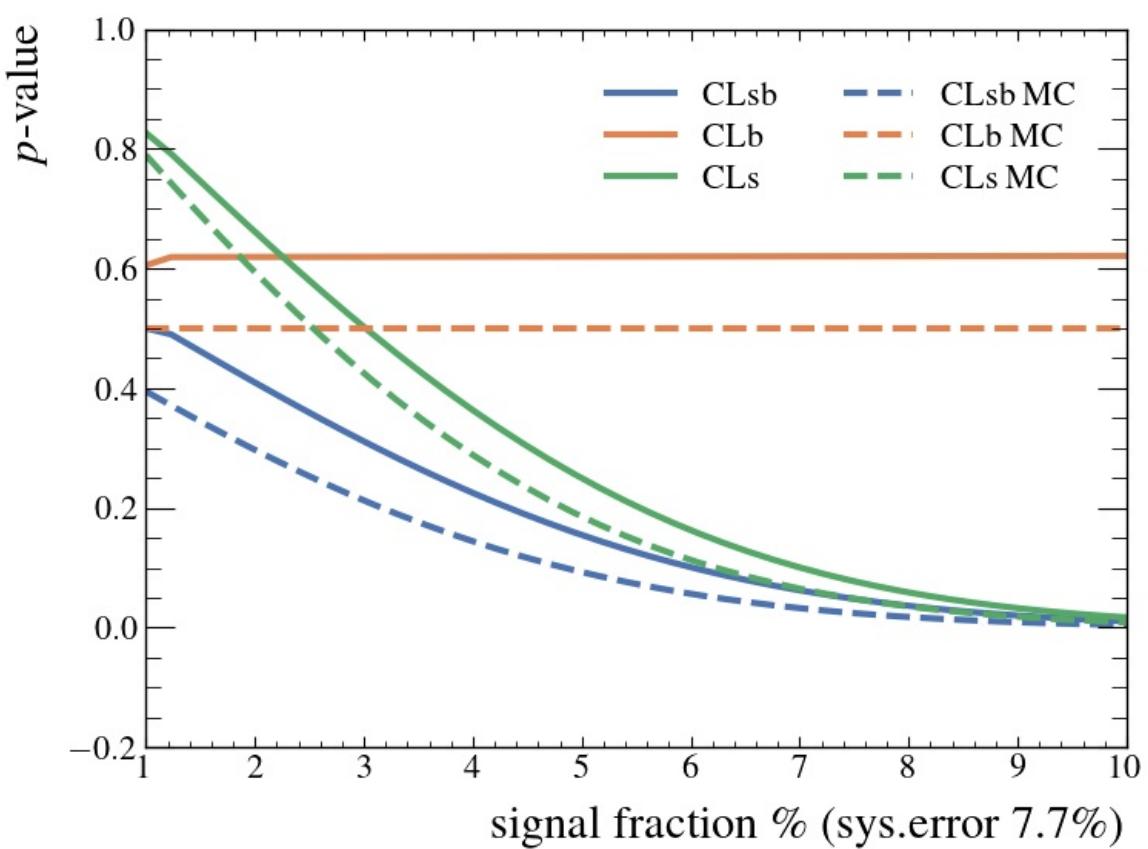


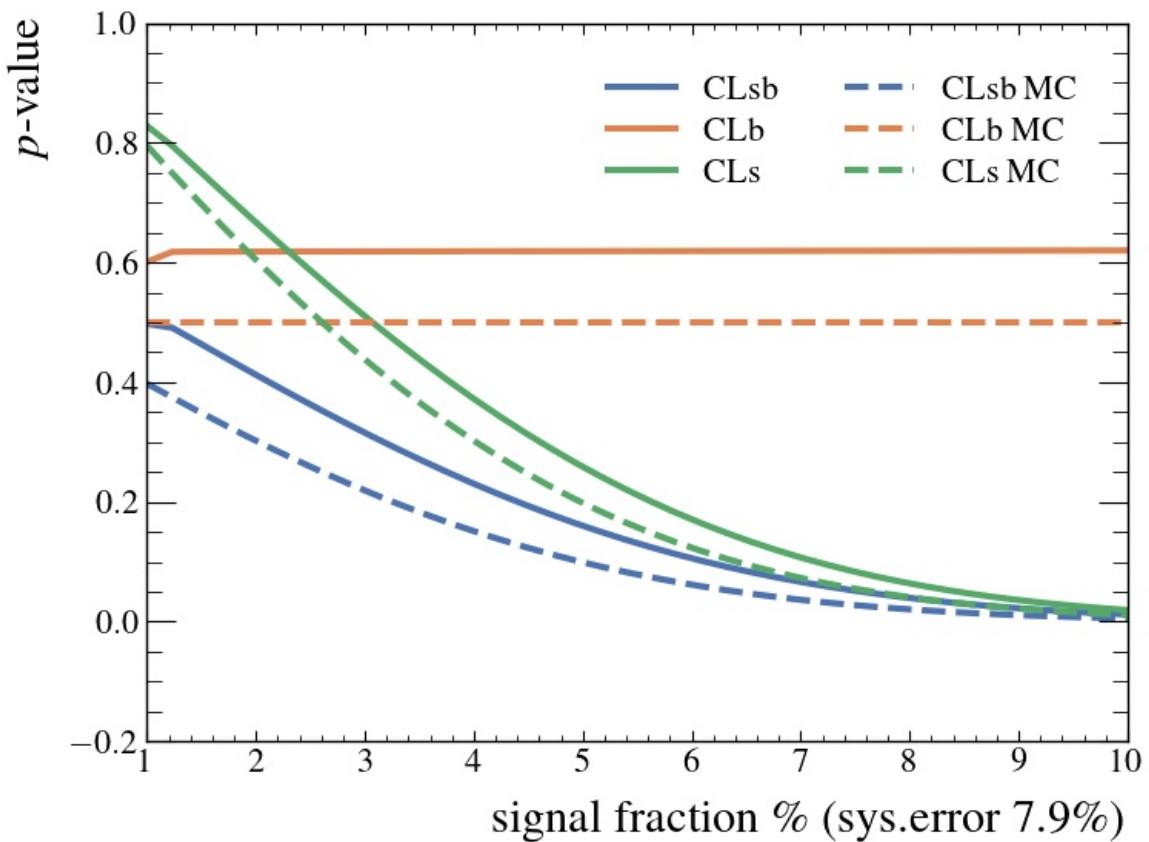


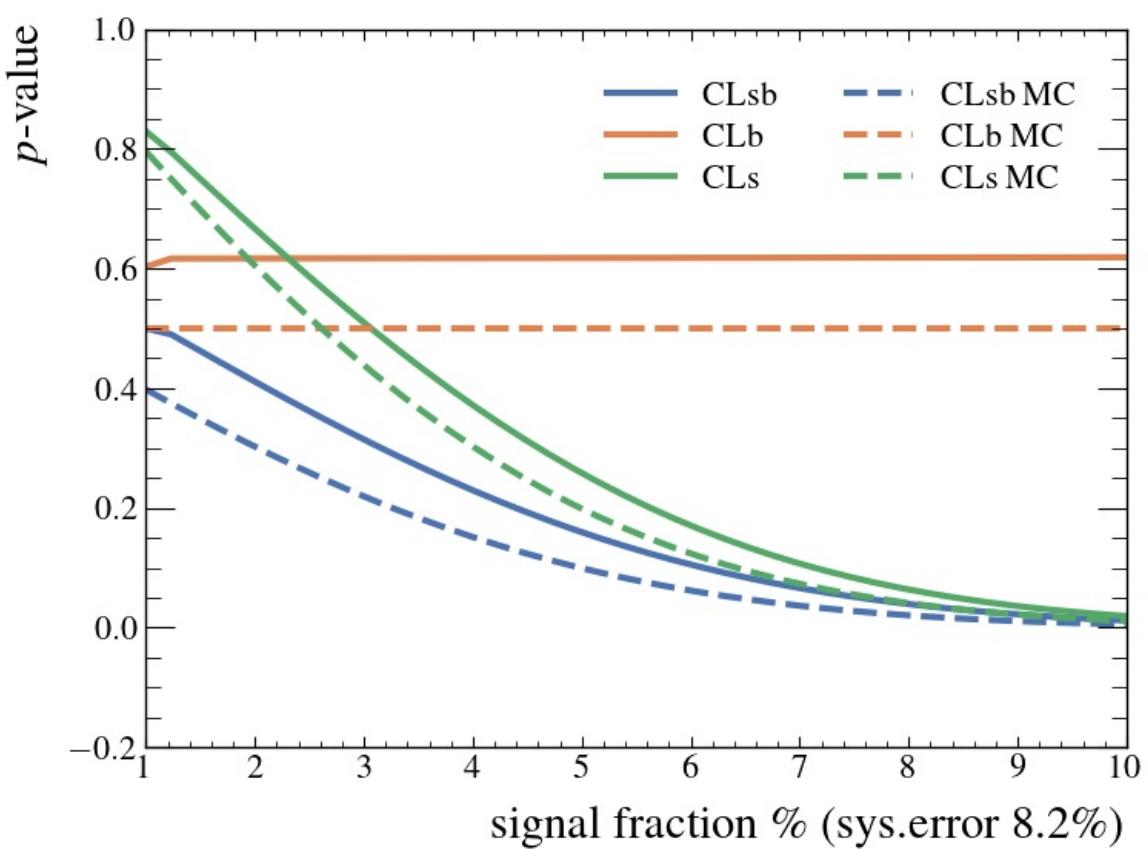


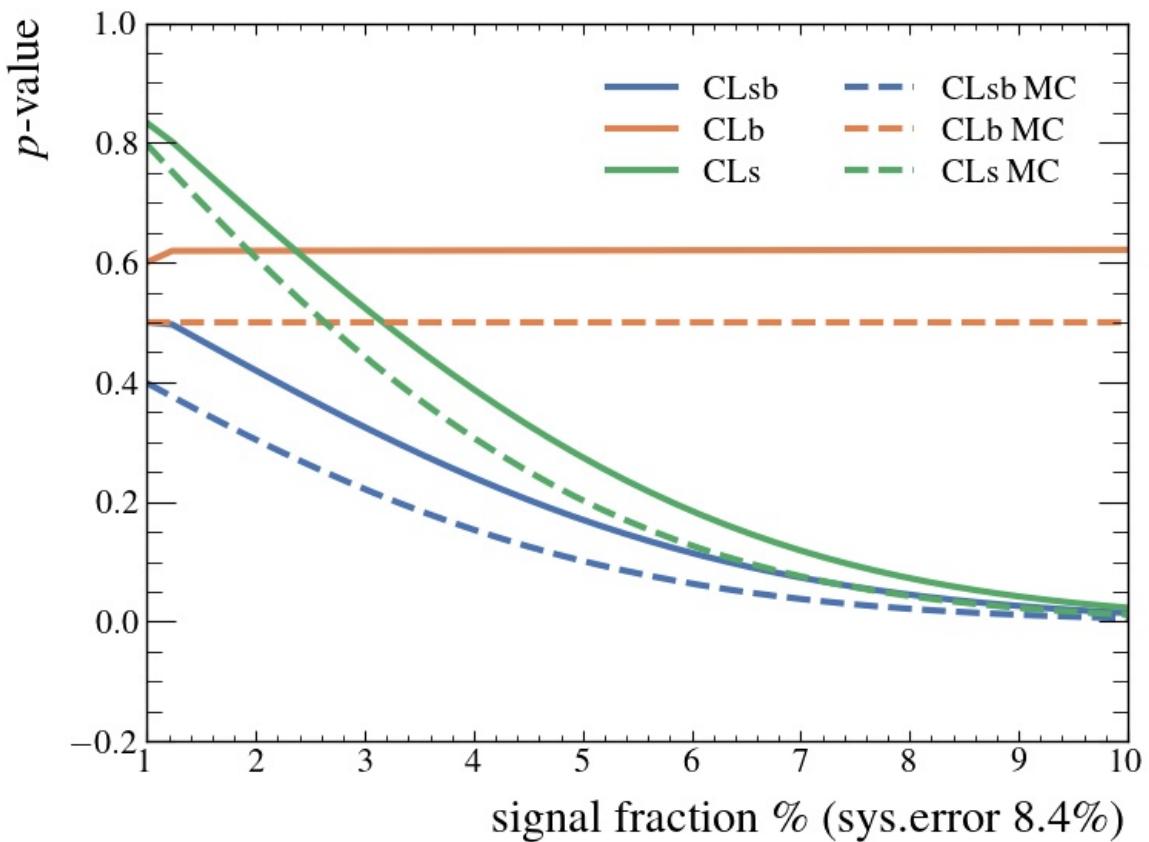


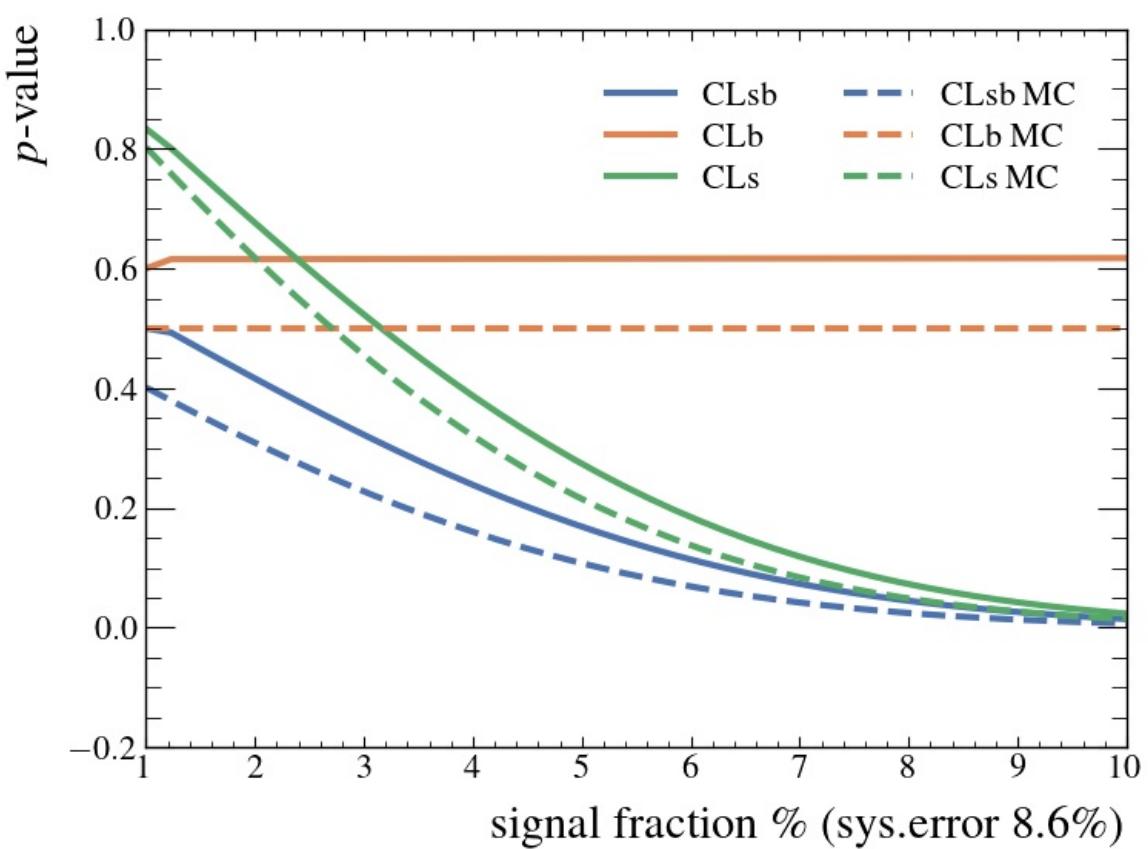


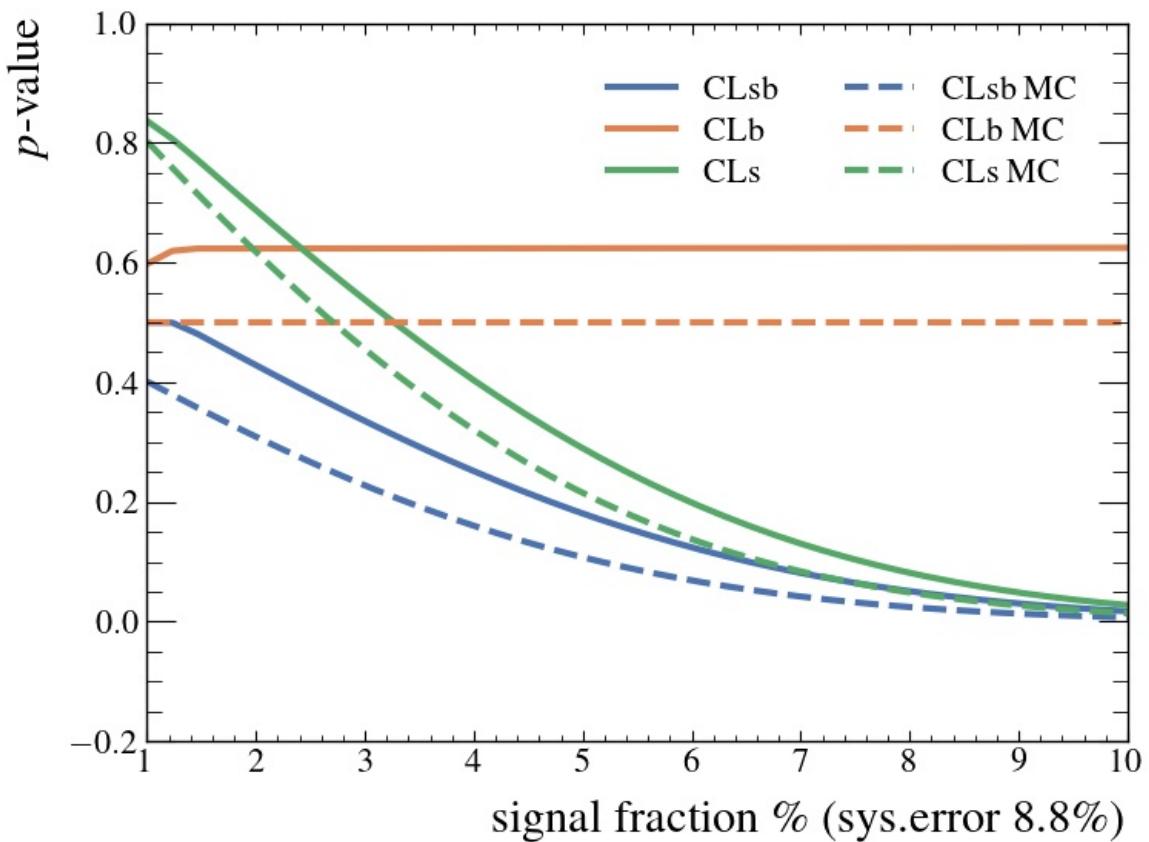


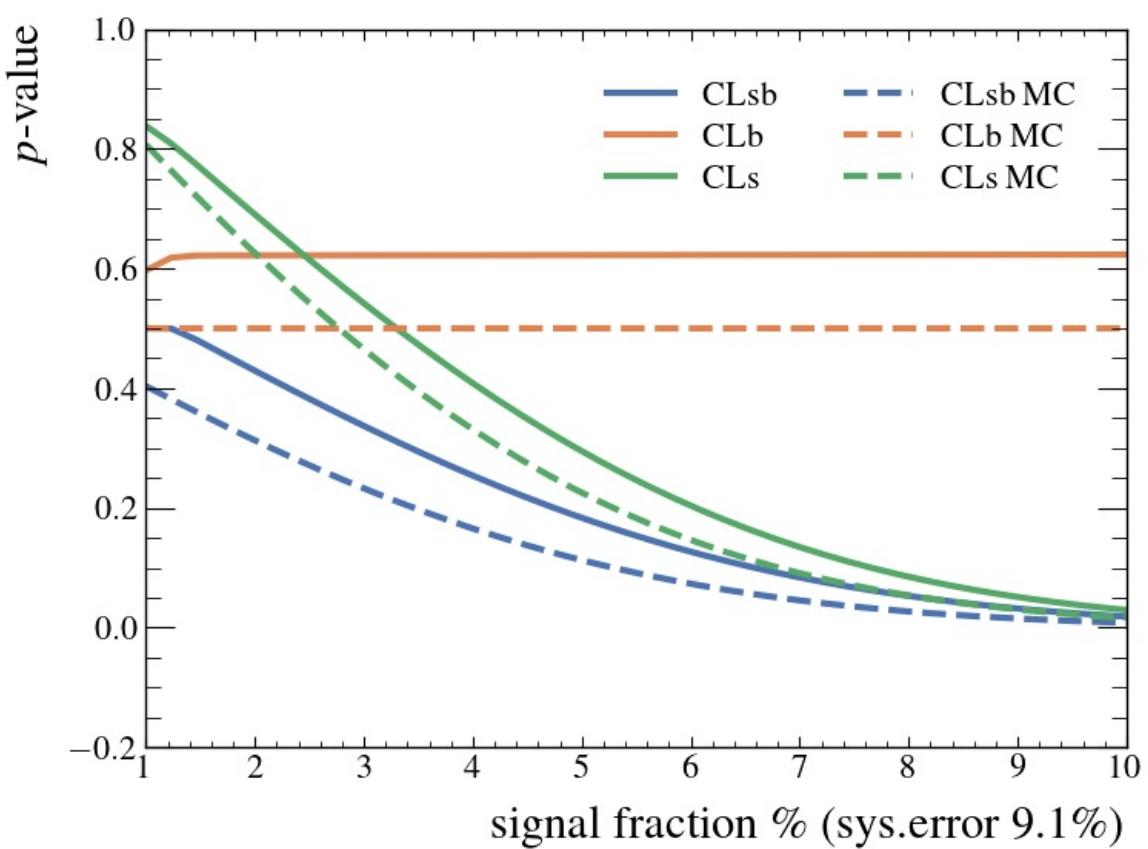


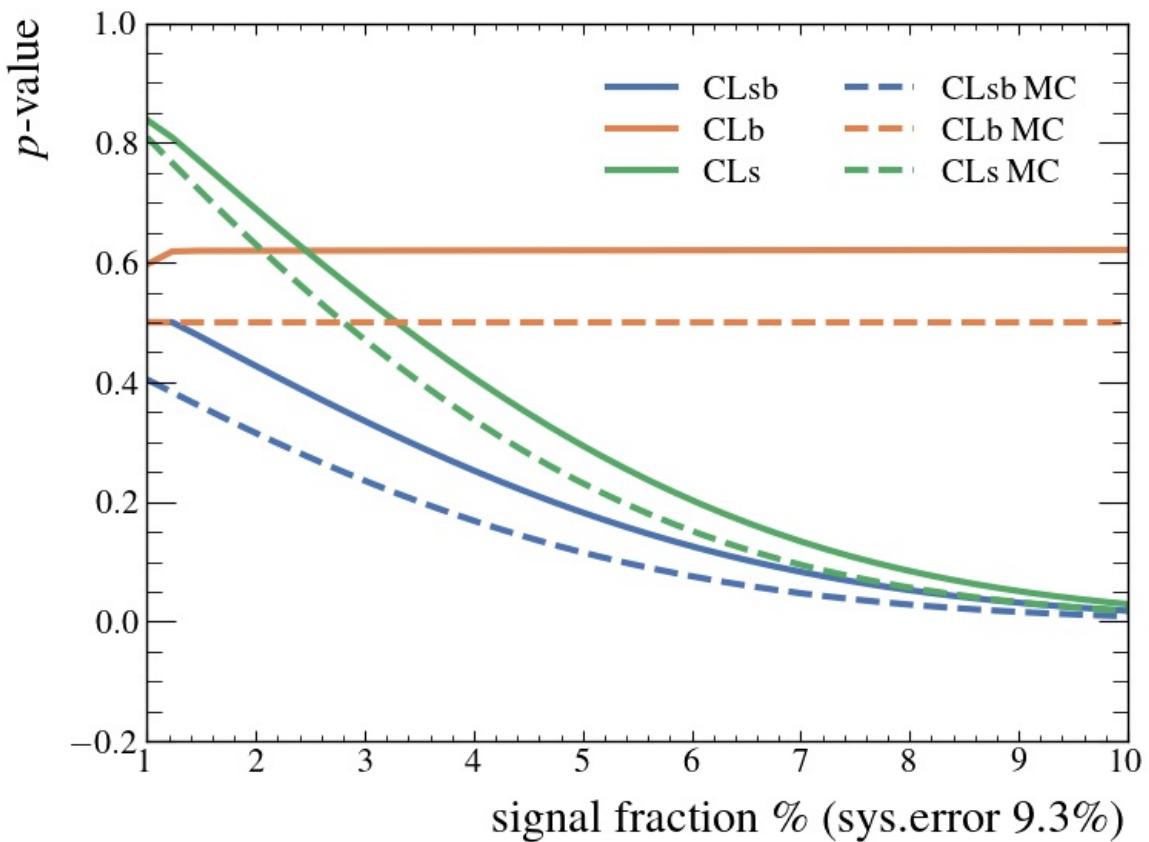


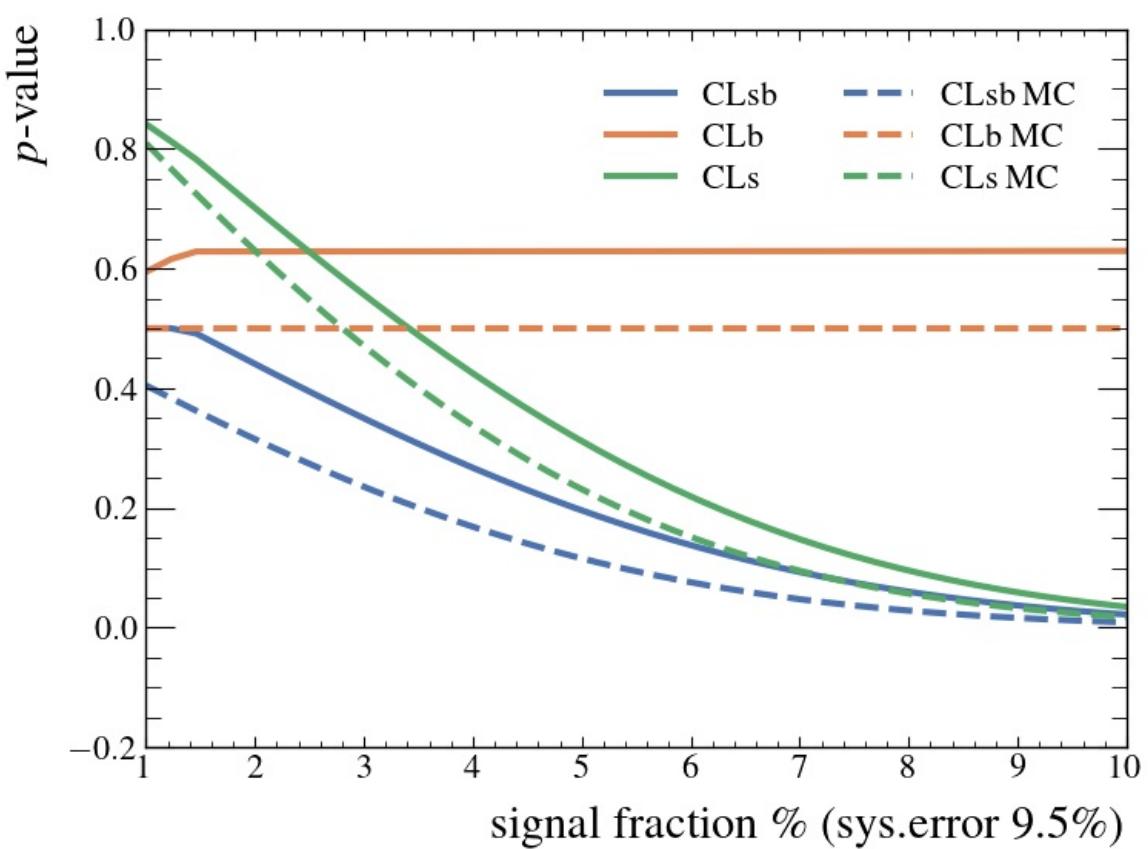


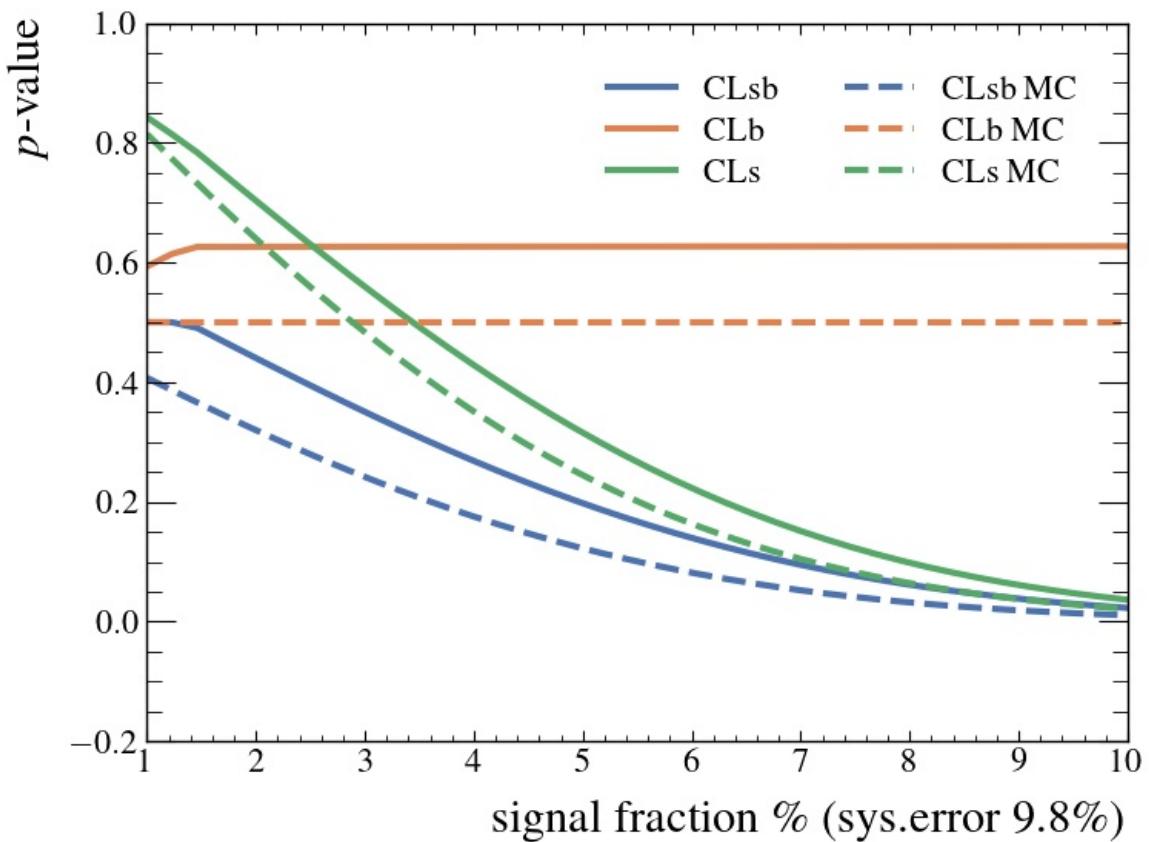


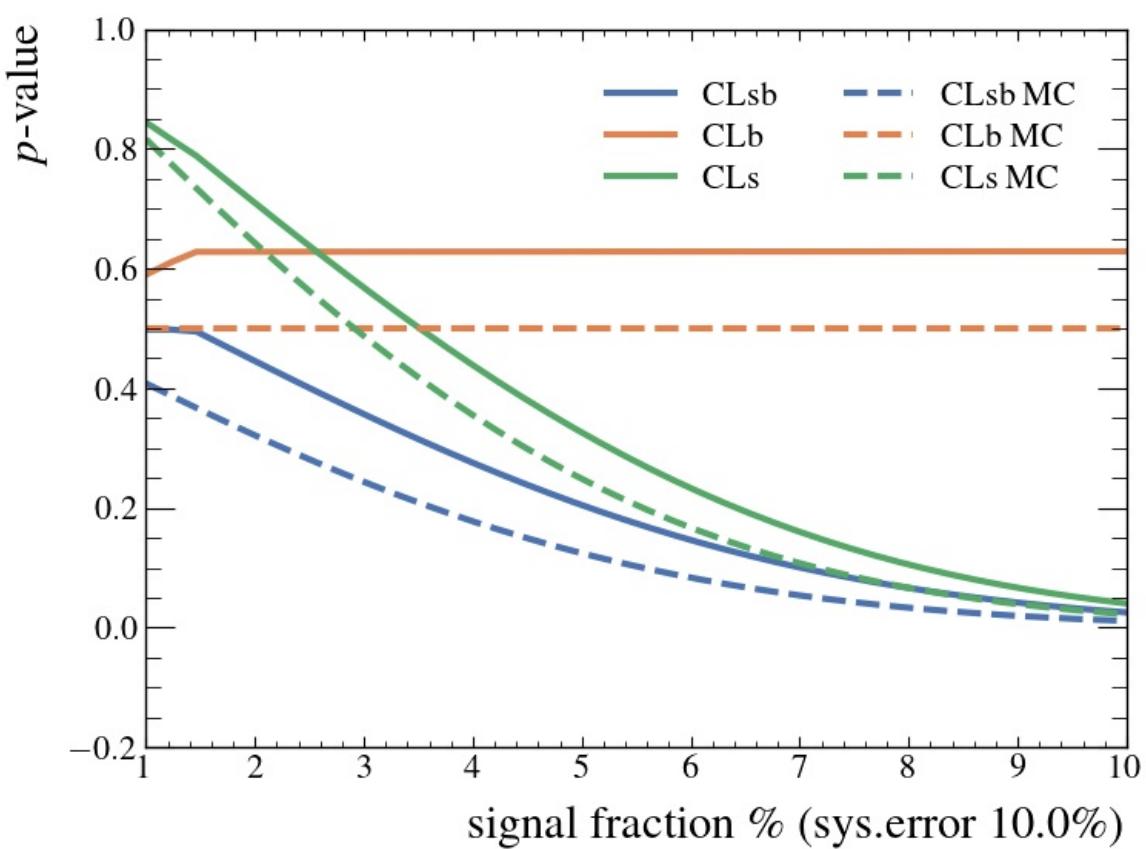












In []:

In []: