Status report

July 1, 2022

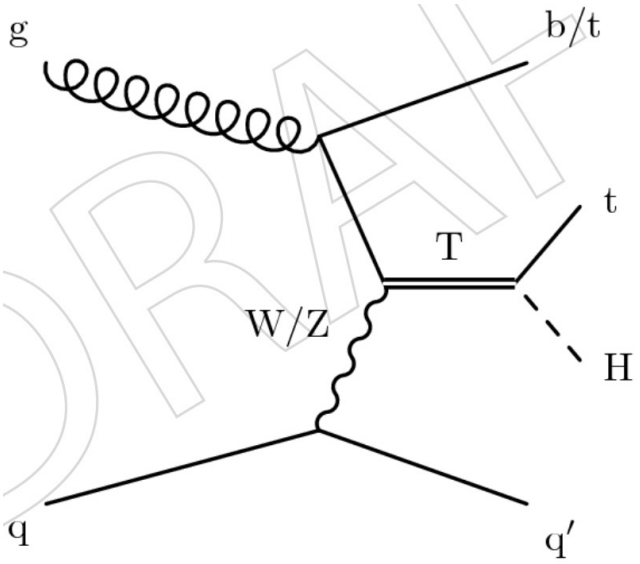
Jieun Choi

HYU / IP2I

Ana development: Take replacement

Track down decay chain (reminder)

- An algorithm was implemented to specify hadronic decay channel from inclusive sample



- Strategy
  - Track any W decays into qq
  - Track any H decays into bb
  - If an event contains hadronic W && Hbb : Hadronic T'

- Expected BR for T' hadronic(t → Wb ~ 1, W→qq ~ 0.66, H→bb ~0.6) : ~ 0.396

- After selection (for a file): 27300 → 10754 ~ 0.394 🙂

Ana development: Take replacement

Track down decay chain

- Replace into ROOT::VecOps::Take()

```cpp
#include "ROOT/RDataFrame.hxx"
#include "ROOT/RVec.hxx"

using ints =  ROOT::VecOps::RVec<int>;
using floats =  ROOT::VecOps::RVec<float>;

ints find_element(ints vec, int a){
    ints out;
    for(unsigned int i = 0; i < vec.size(); i++){
        if( vec[i] == a ) out.emplace_back(i);
    }
    return out;
}

bool isHadTprime(floats &p, ints &pdgId, ints &midx){
    ints out;

    int np = p.size();
    bool flag = false;
    bool Whad = false;
    bool Hbb = false;

    // GenParticle loop
    for(size_t i = 0; i < np; i++){
        ints d_idx = find_element(midx,i);
        // is any W decays into hadronically?
        if (abs(pdgId[i]) == 24){
            // looking for the daughter
            for(size_t d = 0; d < d_idx.size(); d++){
                if( abs(pdgId[d_idx[d]]) < 6 ) Whad = true;
            }
            // W would radiate but it should be small
        }
        // is any H decays into bb?
        else if (abs(pdgId[i]) == 25){
            for(size_t d = 0; d < d_idx.size(); d++){
                if( abs(pdgId[d_idx[d]]) == 5 ) Hbb = true;
            }
        }
    }
    if (Whad && Hbb) flag = true;
    return flag;
};

void hadronic(){

    ROOT::RDataFrame df("Events","/Users/jieun/WORK/vlq/source/TprimeBToTH_M-700_LH_2018/nanoTree_1.root");

    auto base = df.Define("flag",isHadTprime,{"GenPart_pt","GenPart_pdgId","GenPart_genPartIdxMother"})
                  .Filter("flag"); // event selections having hadronic decay

    auto n_df = df.Count();
    auto n_df_S1 = base.Count();

    std::cout << "All counts " << *n_df <<  std::endl;
    std::cout << "Hadronic Tprime decay " << *n_df_S1 << std::endl;
```

5.707 s

→

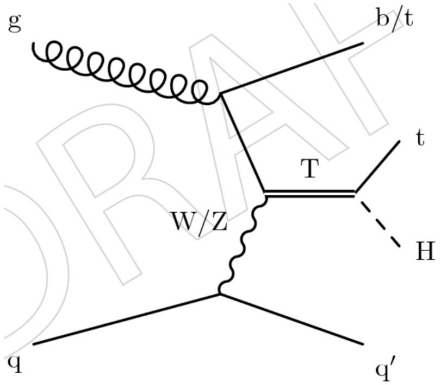
```cpp
#include "ROOT/RDataFrame.hxx"
#include "ROOT/RVec.hxx"

void takeEasy_sum()
{
    ROOT::RDataFrame df("Events","/Users/jieun/WORK/vlq/source/TprimeBToTH_M-700_LH_2018/nanoTree_1.root");

    // looking for W->hadronic(pdgId <9) and H->bb decays in Gen-level
    auto launcher = df.Define("isGenQ","abs(GenPart_pdgId)<9")
                      .Define("GenQmom","GenPart_genPartIdxMother[isGenQ]")
                      .Define("GenQmom_pdgId","Take(GenPart_pdgId,GenQmom)")
                      .Define("isQFromW","abs(GenQmom_pdgId)==24")

                      .Define("isGenb","abs(GenPart_pdgId)==5")
                      .Define("Genbmom","GenPart_genPartIdxMother[isGenb]")
                      .Define("Genbmom_pdgId","Take(GenPart_pdgId,Genbmom)")
                      .Define("isbFromH","abs(Genbmom_pdgId)==25");

    //in gen-level it should be two
    auto sum = launcher.Filter("Sum(isQFromW) > 1 && Sum(isbFromH) > 1");

    auto n_df = launcher.Count();
    auto n_df_S1 = sum.Count();

    std::cout << "All counts " << *n_df <<  std::endl;
    std::cout << "Hadronic Tprime decay " << *n_df_S1 << std::endl;
}
```

4.397 s

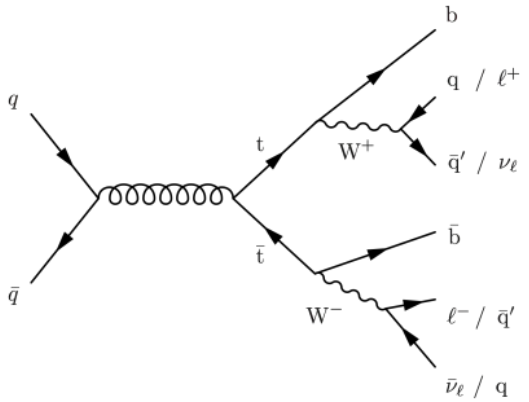
Elapsed time and length are improved
Replacing loops into VecOps functions..

NN in application

Target process



TprimeBToTH_M-700



TTToHadronic

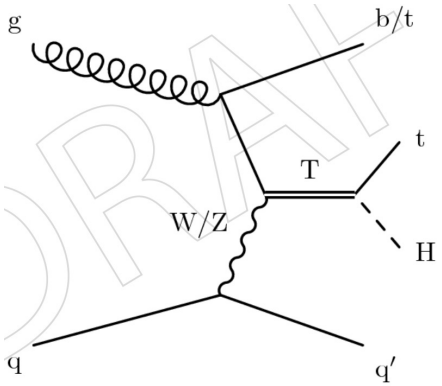
Can we check the significance of cuts via NN?

Basic Selection Criteria	Signal (M=700 GeV/c^2)	QCD_HT_700_1000	$t\bar{t}$
Trigger and p_T,η and $n_b^{DeepCSVT} \geq 3$	4242	3326	83816
$j^1_{p_T} > 170$ GeV/c, $j^2_{p_T} > 130$ GeV/c, $j^3_{p_T} > 80$ GeV/c and $H_T > 500$ GeV/c	2577	1964	28779
$\chi^2 < 15$	933	407	10251
2nd Top Mass> 250 GeV/c^2	855	390	8617
Higgs Mass> 100 GeV/c^2	795	328	7492

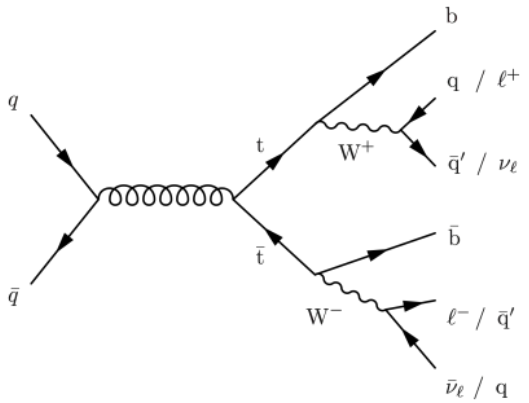
year	2018UL						
Label	Cuts	Data	Signal M=600 GeV/c^2	Signal M=700 GeV/c^2	Signal M=900 GeV/c^2	Signal M=1100 GeV/c^2	$t\bar{t}$
Cut 0	Basic selection	23518	50.6 ± 1.1	41.8 ± 0.7	16.4 ± 0.2	5.0 ± 0.1	4459.0 ± 31.1
Cut 1	Relative $H_T > 0.4$	11505	38.5 ± 0.9	35.7 ± 0.7	14.7 ± 0.2	4.5 ± 0.1	2396.4 ± 23.0
Cut 2	Max(χ^2) < 3	4444	23.1 ± 0.7	23.8 ± 0.5	9.9 ± 0.2	3.0 ± 0.1	1234.9 ± 16.3
Cut 3	$\Delta R(b_{Higgs}, b_{Higgs}) < 1.1$	2328	15.5 ± 0.6	19.8 ± 0.5	9.1 ± 0.2	2.8 ± 0.1	595.0 ± 11.5
Cut 4	$\chi^2_{Higgs} < 1.5$	1961	13.6 ± 0.6	17.6 ± 0.5	8.2 ± 0.2	2.5 ± 0.1	492.8 ± 10.5
Cut 5	$\Delta R(j_W, j_W) < 1.75$	1315	10.6 ± 0.5	15.2 ± 0.4	7.3 ± 0.2	2.3 ± 0.1	389.9 ± 9.3
Cut 6	$\Delta R(b_{Top}, W) < 1.2$	593	6.1 ± 0.4	11.4 ± 0.4	6.4 ± 0.1	2.0 ± 0.1	250.8 ± 7.5

NN in application

Target process



TprimeBToTH_M-700



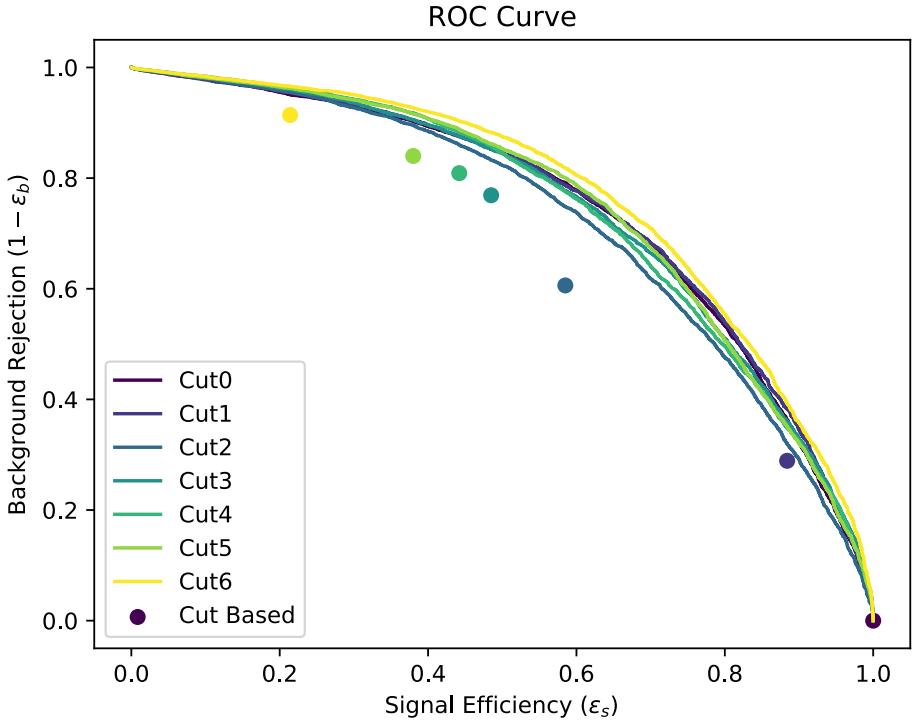
TTToHadronic

Check the importance of cuts via NN

- Variables which were used for Cut0
  - {eta, energy/chi2mass, b-disc.} of (b-tagged) jets, goodHT, Chi2_min, 2nd Top Mass, Higgs Mass
  - njets, nbjets
- Cut1: RelHT
- Cut2: Chi2_Max
- Cut3: Chi2_dRHbb
- Cut4: Chi2_min_H
- Cut5: Chi2_dRWjj
- Cut6: Chi2_dRbW
- Candidates (not tested yet): Chi2_min_W, Chi2_min_Top, mindR_dRbb, mindR_mbb, w_mass, top_mass, chi2_mass, w+H_mass, mass(top-H / top+H), mass(2ndTop+2ndW / H), Chi2_dRHTop, Chi2_dEtaWH, Chi2_dPhiHTop, pT (2ndTop – top / 2ndTop), pT (H-top / Tprime), pT (T'/H - T'/top)

NN in step by step

DNN Structure:
3 layers with 100 nodes
Dropout: 0.1
Activation: relu+sigmoid
Optimizer: Adam
Loss: binary_crossentropy
Batch size: 2048



Cut step	*AUC (1)*	*AUC (2)*	*AUC (3)*
Cut 0	0.7520	0.7462	0.7588
Cut 1	0.7544	0.7500	0.7351
Cut 2	0.7418	0.7236	0.7222
Cut 3	0.7530	0.7446	0.7316
Cut 4	0.7627	0.7419	0.7559
Cut 5	0.7624	0.7488	0.7671
Cut 6	0.7648	0.7693	0.7566

AUC: Area Under Curve

Details

- Training strategy is the same: (with TprimeBToTH_M-700 2018)
  - Train on odd numbered event after DNN selection (HLT+6 jets+3 bjets)
  - Evaluate on even numbered event after cut 0
- Compared performances by adding input variables used in cutBased without cutting itself
- Performance fluctuates ~ AUC $\pm$ 0.02 (*tested for 3 trainings per each cuts (Table 1~3)*)
  - Once a model is determined, performance does not change
  - Not much differences observed from adding a few variables
  - General performance tend to go up by adding cut 0 to cut 6, but drop down at Cut2 (Chi2_Max)