Truth level input for learning with dark Higgs model (bug-fixed)

Philipp Gadow (DESY)
19.04.2021

Motivation

Active learning R&D benefits from iterations with quick feedback on which model configuration is excluded and which still is viable:

This is provided by a truth-based implementation of the ETmiss + h(bb) search.

- Truth-based implementation neglects time-consuming detector simulation.
- Using a HTCondor batch system for parallel execution, a grid of 42 points can be evaluated in less than three hours, including signal generation, TRUTH derivation production, running the truth implementation and estimating limits
- The truth implementation has limited precision compared to the full RECAST but can be useful for prototyping or for exploration of the parameter space.

Workflow

https://github.com/philippgadow/active_learning_darkhiggs

signal generation

generate **events** and compute **cross-section** using ATLAS MC production software

input parameters:
Z' boson mass
dark Higgs boson mass
dark matter particle mass
coupling gx
coupling gs

truth-level analysis

evaluate **product of detector acceptance and selection efficiency** using truth-level implementation of analysis selection in SimpleAnalysis

limits / sensitivity estimate

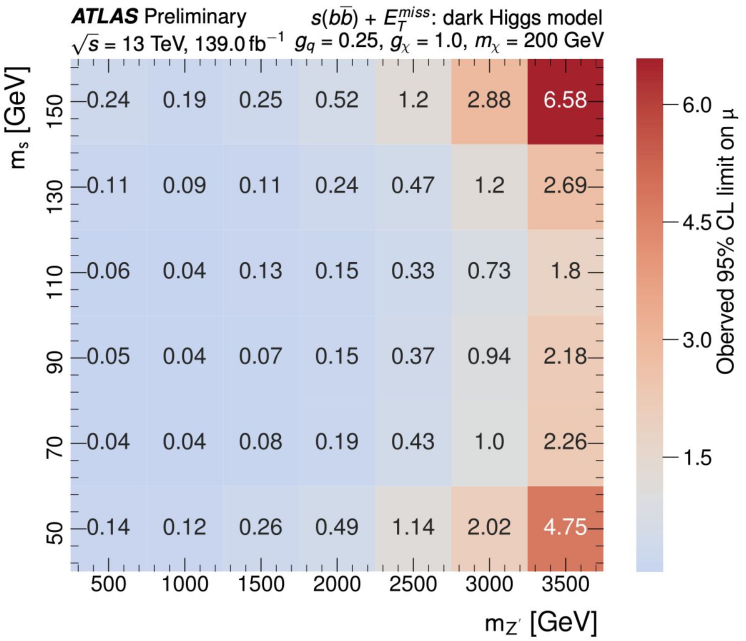
estimate **sensitivity** based on exclusion limits on E_T^{miss} + h(bb) cross-section provided by full Run-2 search

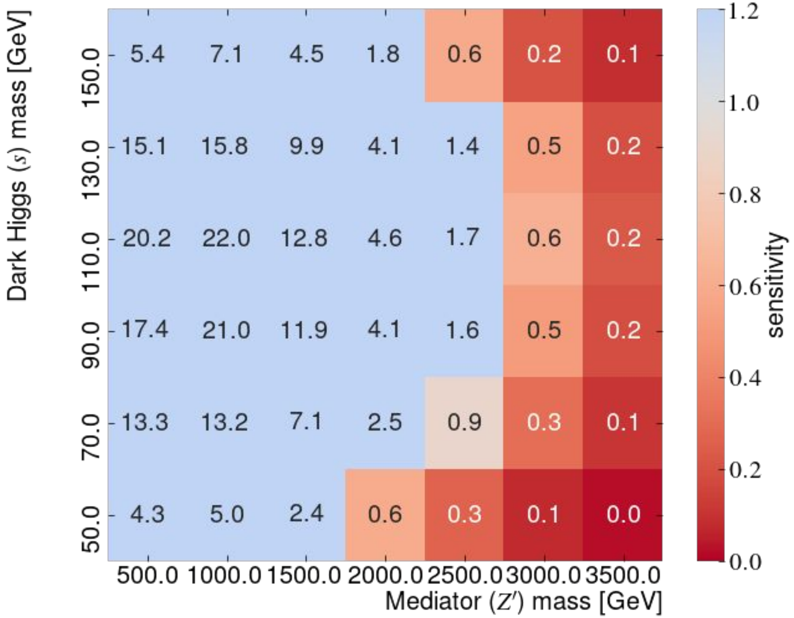
result:
“sensitivity”

if sensitivity is larger than 1, the signal point is expected to be excluded

Exclusion heatmap

Full Run 2 MonoH RECAST



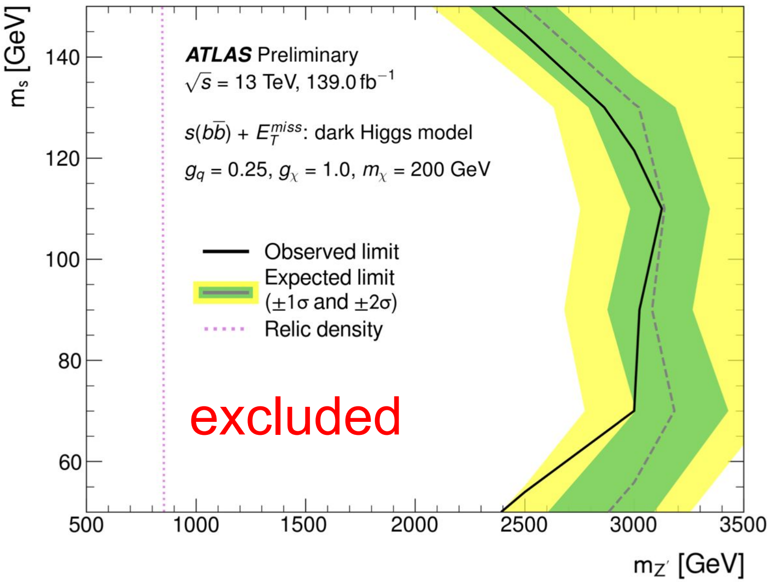


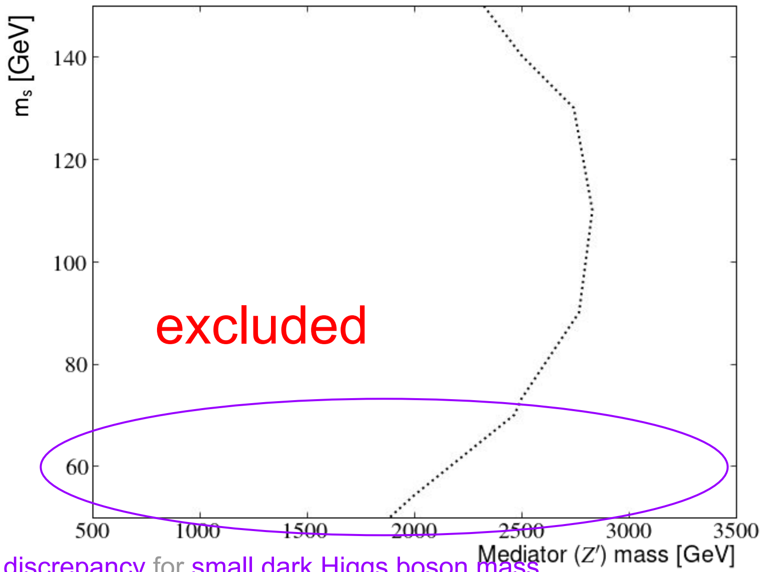
[Proper RECAST (E. Skorda)]

Estimate using SimpleAnalysis truth level implementation and E_T^{miss} + h(bb) limits

Exclusion contour

Full Run 2 MonoH RECAST





some discrepancy for small dark Higgs boson mass
(see discussion on next slide)

[Proper RECAST (E. Skorda)]

Estimate using SimpleAnalysis truth level implementation and E_T^{miss} + h(bb) limits

Drawbacks

- simplified description of analysis on truth level
    - e.g. complicated selection requirements such as extended tau veto had to be neglected
    - currently, no smearing is applied
- sensitivity estimate is based on E_T^{miss} + h(bb) cross-section limits
    - therefore, implicit assumption that Higgs mass equals 125 GeV
    - explains large discrepancies for small dark Higgs boson mass when comparing RECAST to truth level implementation

Potential improvements

- could consider pyhf likelihood instead of cross-section limits
    - thus, avoiding the implicit assumption which is clearly wrong for low dark Higgs masses

additional material

How does this sensitivity computation work?

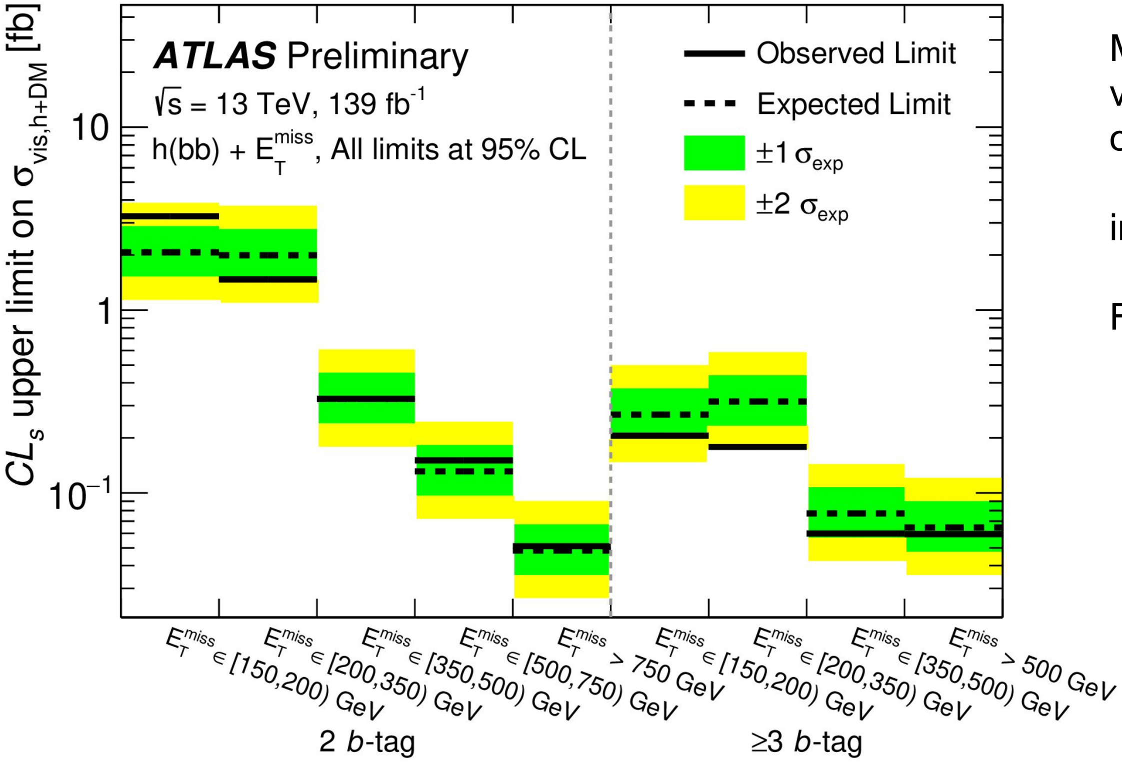
Original idea proposed by Oleg Brandt and his students (see talk by S. Suchek).

1. Compute cross-section for signal point
2. Calculate product of acceptance and efficiency for signal point on truth-level, neglecting detector effects for each bin defined in E_T^{miss} and b-tag multiplicity.
3. In each bin, compute

$$S_i = \frac{\sigma\ \left(pp \to s + E_T^{\mathrm{miss}}\right)_{\textit{dark Higgs}} \cdot \mathrm{BR}\left(s \to b\bar{b}\right)_{\textit{dark Higgs}} \cdot (\mathcal{A} \cdot \varepsilon)_i}{\sigma_i\left(pp \to h + E_T^{\mathrm{miss}} \to b\bar{b} + E_T^{\mathrm{miss}}\right)_{\mathrm{obs}}}$$

4. The total sensitivity is the sum over all S_i in all bins (which are denoted by *i*)

Cross-section limits on E_T^{miss} + h(bb)



Model-independent upper limits on the visible cross-section

$\sigma_{vis,h(b\bar{b})+DM} \equiv \sigma_{h+DM} \times B(h \to b\bar{b}) \times A \times \epsilon$

in the different signal regions.

Figure from ATLAS-CONF-2021-006

SimpleAnalysis implementation

- consider TRUTH1 derivations as inputs which include large-radius jets based on TruthParticles and variable-radius track jets based on charged truth particles
- truth object definition similar as in reco-level E_T^{miss} + h(bb) search

SimpleAnalysis object definition 1/3

```
// Object definitions
// base leptons (used in veto definitions)
auto baselineElectrons = event->getElectrons(7, 2.47, ELooseBLLH | EIsoFixedCutLoose);
auto baselineMuons = event->getMuons(7, 2.5, MuLoose | MuIsoFixedCutLoose);
auto baselineTaus = event->getTaus(20, 2.5, TauRNNVeryLoose);
// central jets: pt > 20 GeV, |eta| < 2.5
auto centralJets = event->getJets(20., 2.5, PFlowJet | JVTMedium);
// forward jets: pt > 30 GeV, 2.5 < |eta| < 4.5 -> need to get creative with "filterCrack"
auto forwardJets = event->getJets(30., 4.5);
forwardJets = filterCrack(forwardJets, 2.5, 4.5);
sortObjectsByPt(centralJets);
sortObjectsByPt(forwardJets);
auto allJets = centralJets + forwardJets;
auto fatJets = event->getFatJets(200., 2.0);
auto trackJets = event->getTrackJets(10., 2.5);
```

SimpleAnalysis object definition 2/3

```
// Overlap removal - including with object Pt-dependent radius calculation
auto radiusCalcJet = [](const AnalysisObject &, const AnalysisObject &muon) {
  return std::min(0.4, 0.04 + 10 / muon.Pt());
};
auto radiusCalcMuon = [](const AnalysisObject &muon, const AnalysisObject &) {
  return std::min(0.4, 0.04 + 10 / muon.Pt());
};
baselineTaus = overlapRemoval(baselineTaus, baselineElectrons, 0.2);
baselineElectrons = overlapRemoval(baselineElectrons, baselineMuons, 0.01);
centralJets = overlapRemoval(centralJets, baselineElectrons, 0.2, NOT(BTag77DL1));
baselineElectrons = overlapRemoval(baselineElectrons, centralJets, 0.4);
centralJets = overlapRemoval(centralJets, baselineMuons, radiusCalcJet, LessThan3Tracks);
baselineMuons = overlapRemoval(baselineMuons, centralJets, radiusCalcMuon);
centralJets = overlapRemoval(centralJets, baselineTaus, 0.2);
fatJets = overlapRemoval(fatJets, baselineElectrons, 1.0);
```

SimpleAnalysis object definition 3/3

```
// Advanced object definitions
auto bjets = filterObjects(centralJets, 20., 2.5, BTag77DL1);
auto btrackjets = filterObjects(trackJets, 20., 2.5, BTag77DL1);
auto baselineLeptons = baselineElectrons + baselineMuons;
sortObjectsByPt(bjets);
sortObjectsByPt(btrackjets);
sortObjectsByPt(centralJets);
sortObjectsByPt(fatJets);
sortObjectsByPt(baselineLeptons);
```

here was a bug: confused centralJets and trackJets, now fixed

SimpleAnalysis event reconstruction 1/2

```
// Event-level observable reconstruction
int nBjets = bjets.size();
float dphiMin3 = minDphi(metVec, allJets, 3);
float mt_min = 0;
float mt_max = 0;
float mindR = 999.;
float maxdR = -1;
 for (const auto &jet : bjets) {
   if (metVec.DeltaR(jet) < mindR) {
     mindR = metVec.DeltaR(jet);
     mt_min = calcMT(jet, metVec);
   }
   if (metVec.DeltaR(jet) > maxdR) {
     maxdR = metVec.DeltaR(jet);
     mt_max = calcMT(jet, metVec);
   }
}
```

SimpleAnalysis event reconstruction 2/2

```
// Higgs candidate reconstruction
 float mHiggs = 0.;
 float ptHiggs = 0.;
 int nBJetsMerged = 0;
 if (met > 500) {
   if (fatJets.size() > 0) {
     mHiggs = fatJets[0].M();
     ptHiggs = fatJets[0].Pt();
     for (const auto &jet : btrackjets) {
       if (jet.DeltaR(fatJets[0]) > 1.) continue;
       nBJetsMerged++;
     }
   }
 } else {
   if (nBjets >=2) {
     mHiggs = (bjets[0]+bjets[1]).M();
     ptHiggs = (bjets[0]+bjets[1]).Pt();
   }
 }
```

SimpleAnalysis event selection 1/2

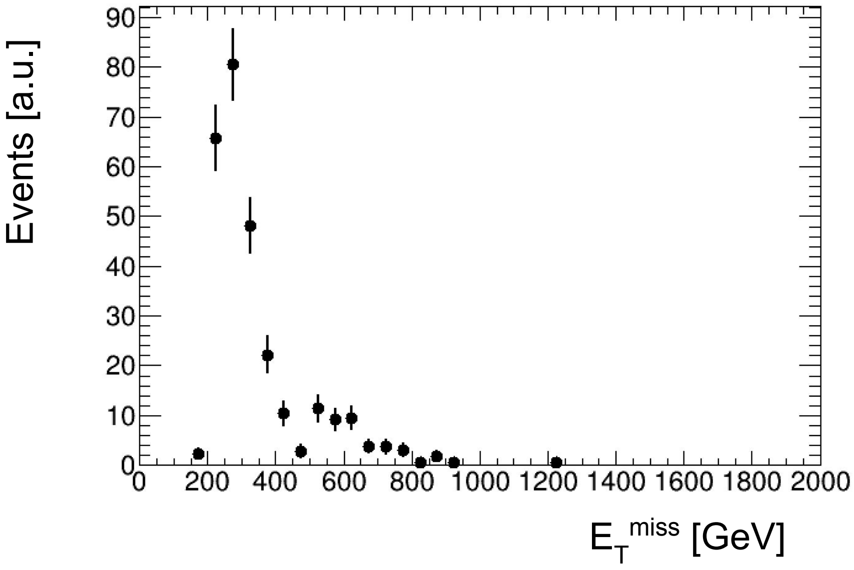
```
// common cuts in merged and resolved event selections
 bool evtsel_met150 = (met > 150.);
 bool evtsel_leptonVeto = (baselineLeptons.size() == 0);
 bool evtsel_tauVeto = (baselineTaus.size() == 0);
 bool evtsel_extendedTauVeto = true; // TODO: implement
 bool evtsel_minDPhi20 = (dphiMin3 > 20. * M_PI / 180.);
 // cuts in merged selection
 bool evtsel_met500 = (met > 500.);
 bool evtsel_massRange_merged = (mHiggs > 50. && mHiggs < 270.);
 // cuts in resolved selection
 bool evtsel_metleq500 = (met <= 500.);
 bool evtsel_njets = (centralJets.size() >=2);
 bool evtsel_nbjets = (nBjets >=2);
 bool evtsel_ptHiggs = ((met <= 350. && ptHiggs > 100) || (met > 350. && ptHiggs > 300));
 bool evtsel_mt_mindR = (mt_min > 170.);
 bool evtsel_mt_maxdR = (mt_max > 200.);
 bool evtsel_metSig = (metSignificance > 12.);
 bool evtsel_njets_max = ((nBjets == 2 && centralJets.size() <= 4) || (nBjets >=3 && centralJets.size() <= 5));
 bool evtsel_massRange_resolved = (mHiggs > 50. && mHiggs < 280.);
```

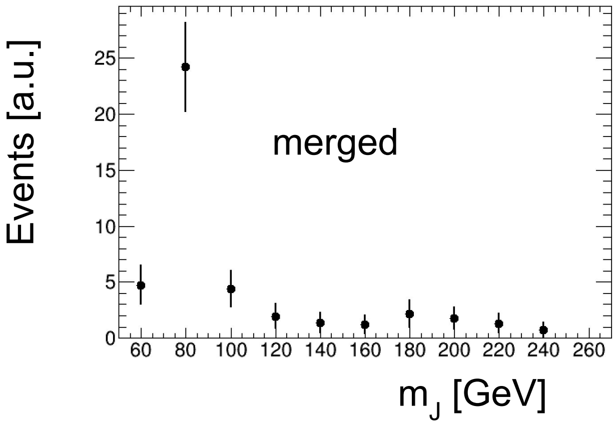
SimpleAnalysis event selection 2/2

```
// Merged selection
bool passMerged = evtsel_met500 && evtsel_massRange_merged;
 // Resolved selection
bool passResolved = evtsel_metleq500 && evtsel_njets && evtsel_nbjets && evtsel_ptHiggs && \
                evtsel_mt_mindR && evtsel_mt_maxdR && evtsel_metSig && evtsel_njets_max && \
                evtsel_massRange_resolved;
if (passMerged) {
   if (nBJetsMerged == 2 && met > 500 && met <= 750) accept("MET500750_2b");
   if (nBJetsMerged == 2 && met > 750)               accept("MET750_2b");
   if (nBJetsMerged >= 3 && met > 500 )              accept("MET500_3b");
 }
 if (passResolved) {
   if (nBjets == 2 && met > 150 && met <= 200) accept("MET150200_2b");
   if (nBjets == 2 && met > 200 && met <= 350) accept("MET200350_2b");
   if (nBjets == 2 && met > 350 && met <= 500) accept("MET350500_2b");

   if (nBjets >= 3 && met > 150 && met <= 200) accept("MET150200_3b");
   if (nBjets >= 3 && met > 200 && met <= 350) accept("MET200350_3b");
   if (nBjets >= 3 && met > 350 && met <= 500) accept("MET350500_3b");
 }
```

Truth-level distributions for $m_{Z'}$=.5 TeV, m_s=90 GeV





Explanation of the structure:

1. E_T^{miss} > 500 GeV: merged selection
2. Higgs p_T requirement for E_T^{miss} < 350 GeV and 350 GeV < E_T^{miss} < 500 GeV different

```
((met <= 350. && ptHiggs > 100) ||
(met > 350. && ptHiggs > 300));
```