

# ADL/CutLang ile Fizik Odaklı YEF Analizi



Aytül Adıgüzel (İstanbul Üniversitesi)  
Feyza Başpehlivan (TOBB ETÜ)  
Demircan Demirbağ (Boğaziçi Üniversitesi)  
Berare Göktürk (Boğaziçi Üniversitesi)  
Hazal Candan Kacar (Boğaziçi Üniversitesi)  
Ahmetcan Sansar (İstanbul Üniversitesi)  
Sezen Sekmen (Kyungpook Nat. U.)  
Burak Şen (ODTÜ)  
Gökhan Ünel (UC Irvine)

[cern.ch/adl](http://cern.ch/adl)

YEFİST 2023  
14 Ekim 2023



## Analiz Betimleme Dili (ADL) & CutLang

Mevcut analiz çerçevelerinde, veri analizleri C++ ve Python gibi genel amaçlı dillere dayanan analiz yazılım çerçeveleri kullanarak yapılırlar:

- Bu çerçevelerde **fizik içeriği ve teknik işlemciler birbiri içine geçmişlerdir** ve birlikte yürütülürler.
- Fizik içeriğinin bu dağınık ve standarttan uzak ifade tarzı analizleri geliştirmeyi, anlamayı, içeriği iletmeyi ve yeni modeller ile yorumlamayı zorlaştırır.
- Farklı çerçevelerde çalışmak ve öğrenmek zaman alır.

Önerdiğimiz alternatif sistem:

- **Fizik algoritmasını tamamen teknik detaylardan ayırır.**
- Veriyle daha doğrudan bir etkileşime izin verir.

**Analiz Betimleme Dili & CutLang** yenilikçi bir analiz sistemi sağlar:

- YEF analizinin fizik içeriğini tanımlayan **alana özgü, bildirimsel** bir dil.
- Analiz ile ilgili işler için dili çalıştırılabilir hale getiren araçlar.



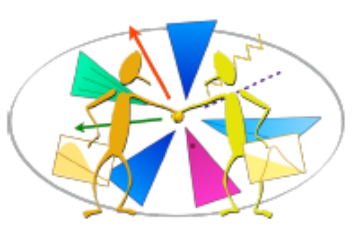
# BHÇ Fiziği için Analiz Betimleme Dili

**Analiz Betimleme Dili (ADL)** bir YEF analizinin fizik içeriğini standart ve açık bir şekilde ifade eden alana özgü, bildirimsel bir dildir (DSL).

- **Harici DSL:** Analize özgü kavramları ifade etmek için özel olarak tasarlanmış sözdizimi. Parçacık fizikçilerinin kavramsal akıl yürütmesini yansıtır. Programlamaya değil fiziğe odaklanır.
- **Bildirimsel:** Yapılması gerekeni açıklar, ancak nasıl yapılacağını tanımlamaz.
- **Kolay okunabilir:** Açık, kendi kendini tanımlayan sözdizimi kuralları.
- **Herkes için tasarlandı:** deneyciler, fenomenoloji yapanlar, öğrenciler, ilgilenen diğer herkese açık...

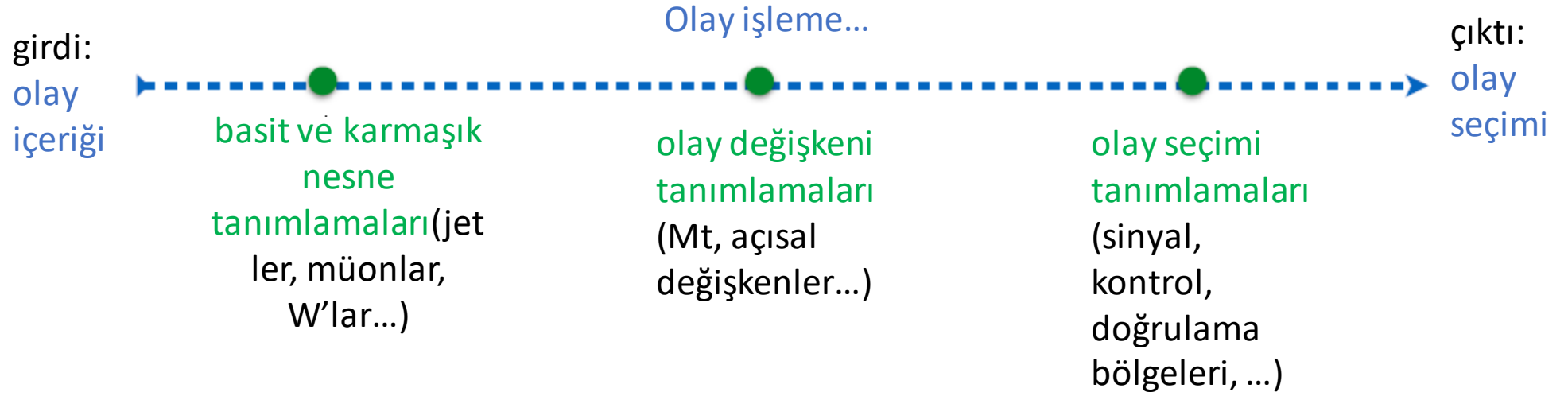
ADL çerçevelerden bağımsızdır --> ADL'yi tanıyan herhangi bir çerçevede analiz koşturulabilir.

- **Çok amaçlı kullanım:** Belirli bir amaç için en uygun GPL / çerçeve otomatik olarak çevrilebilir veya dahil edilebilir. Örneğin deney, analiz, (yeniden) yorumlama, analiz sorguları...
- Gruplar arası **kolay iletişim:** Deneysel, fenomenoloji, hakemler, öğrenciler.
- Analizin fizik içeriğinin **korunumunu sağlar.**



## ADL Kapsamı

- Olay işleme: Öncelikli odak noktamız.



- Analiz sonuçları, örneğin sayımlar ve belirsizlikler: Mevcut.
- Histogramlama: Mevcut.
- Sistemik belirsizlikler: Devam ediyor.
- Seçilen olaylarla işlemler, örneğin aralan tahmini, ölçek faktörü türetme: Henüz kapsam dahilinde değil.



## ADL ile basit bir analiz örneği

### # OBJECTS

```
object goodMuons
```

```
take muon
```

```
select pT(muon) > 20
```

```
select abs(eta(muon)) < 2.4
```

```
object goodEles
```

```
take ele
```

```
select pT(ele) > 20
```

```
select abs(eta(ele)) < 2.5
```

```
object goodLeps
```

```
take union(goodEles, goodMuons)
```

```
object goodJets
```

```
take jet
```

```
select pT(jet) > 30
```

```
select abs(eta(jet)) < 2.4
```

```
reject dR(jet, goodLeps) < 0.4
```

### # EVENT VARIABLES

```
define HT = sum(pT(goodJets))
```

```
define MTI = Sqrt( 2*pT(goodLeps[0]) * MET*(1-cos(phi(METLV[0]) - phi(goodLeps[0]))))
```

### # EVENT SELECTION

```
region baseline
```

```
select size(goodJets) >= 2
```

```
select HT > 200
```

```
select MET / HT <= 1
```

```
region signalregion
```

```
baseline
```

```
select Size(goodLeps) == 0
```

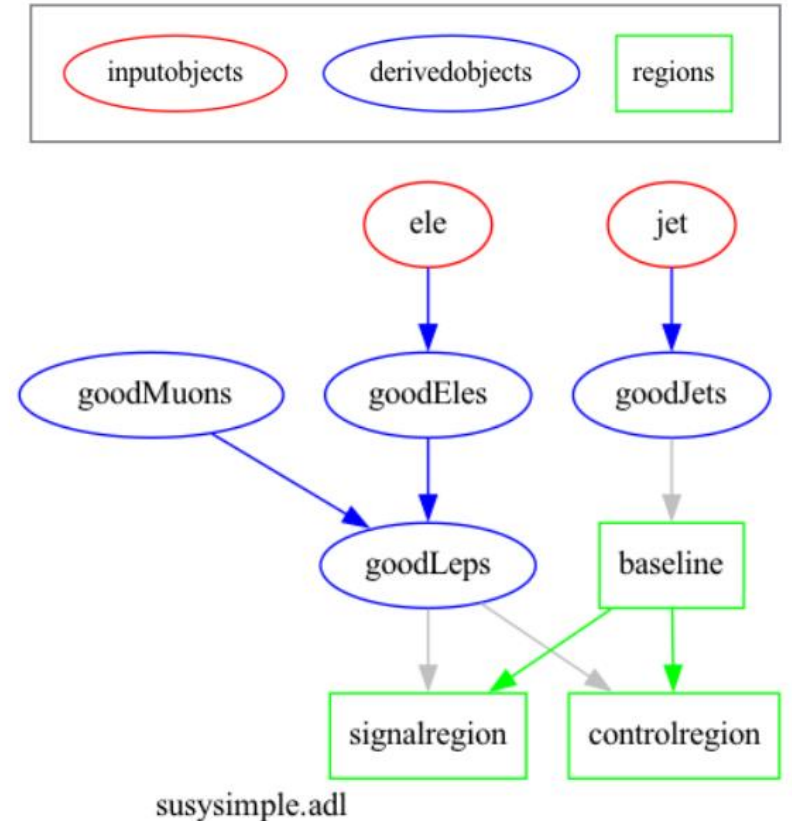
```
select dphi(METLV[0], jets[0]) > 0.5
```

```
region controlregion
```

```
baseline
```

```
select size(goodLeps) == 1
```

```
select MTI < 120
```





# ADL Yapısı

ADL şunlardan oluşur:

- Açık sözdizimi kurallarına sahip okunması kolay bir DSL kullanarak analiz algoritmasını açıklayan bir **düz metin ADL dosyası**.
- ADL sözdizimi ile ifade edilmesi basit olmayan değişkenleri (ör. MT2, ML algoritmaları) içine alan **bağımsız işlevlerden oluşan bir kütüphane**. Dahili veya harici (kullanıcı) fonksiyonları.

- Bir **ADL dosyası** nesne, değişken ve olay seçimi tanımlarını ayıran bloklardan oluşur. Bloklar **anahtar kelime-talimat** yapısına sahiptir.
  - **anahtar kelimeler**, analiz kavramlarını ve işlemlerini belirtir.

```
blocktype blockname  
keyword1 instruction1  
keyword1 instruction2  
keyword3 instruction3 # comment
```

- **Sözdizimi**, matematiksel ve mantıksal işlemleri, karşılaştırma ve optimizasyon operatörlerini, indirgeyicileri, 4 vektör cebirini ve YEF'e özgü fonksiyonları( $d\phi$ ,  $dR$ , ...) içerir.



# CutLang yorumlayıcısı ve çerçeve



CutLang çalışma zamanı yorumlayıcısı:

- Analiz içeriği derlenmez. ADL dosyasını olaylar üzerinden doğrudan çalıştırır.
- CutLang'ın kendisi C++ ile yazılmıştır, herhangi bir modern Unix ortamında çalışır.
- Lorentz Vektör işlemleri ve histogramları için ROOT sınıflarını temel alır.
- ADL sözdizimi işleme: Lex & Yacc gereçleri ile.

CutLang Github: <https://github.com/unelg/CutLang>

CutLang Yayınları: [arXiv:1801.05727](https://arxiv.org/abs/1801.05727), [arXiv:1909.10621](https://arxiv.org/abs/1909.10621)  
[arXiv:2101.09031](https://arxiv.org/abs/2101.09031)

CutLang çerçevesi: yorumlayıcı + araçlar  
Olay ve harici fonksiyon girdileri:

- Olay girdileri: ROOT dosyaları aracılığıyla.
  - Birden çok girdi tipi: Delphes, CMS NanoAOD, ATLAS/CMS Open Data, LVL0, FCC. Daha fazlası kolaylıkla eklenebilir.
  - Tüm olay tipleri çalışma zamanı yorumlayıcısı tarafından tanınacak önceden tanımlanmış parçacık nesne tiplerine dönüştürülür --> aynı ADL dosyası farklı girdi tipleri ile çalıştırılabilir.
  - Yeni Özellik: CutLang, ROOT dosyalarındaki tüm değişkenleri okuyup çalıştırabilir (ilave bilgi sonraki slaytlarda)
- Sonuç çıktıları: ROOT dosyaları aracılığıyla.
  - Her seçim bölgesi için cutflow, binler ve kullanıcı tarafından tanımlanmış histogramların bulunduğu bir TDirectory.



## Güncel altyapı geliřmeleri

CutLang'ın altyapısındaki yenilikler:

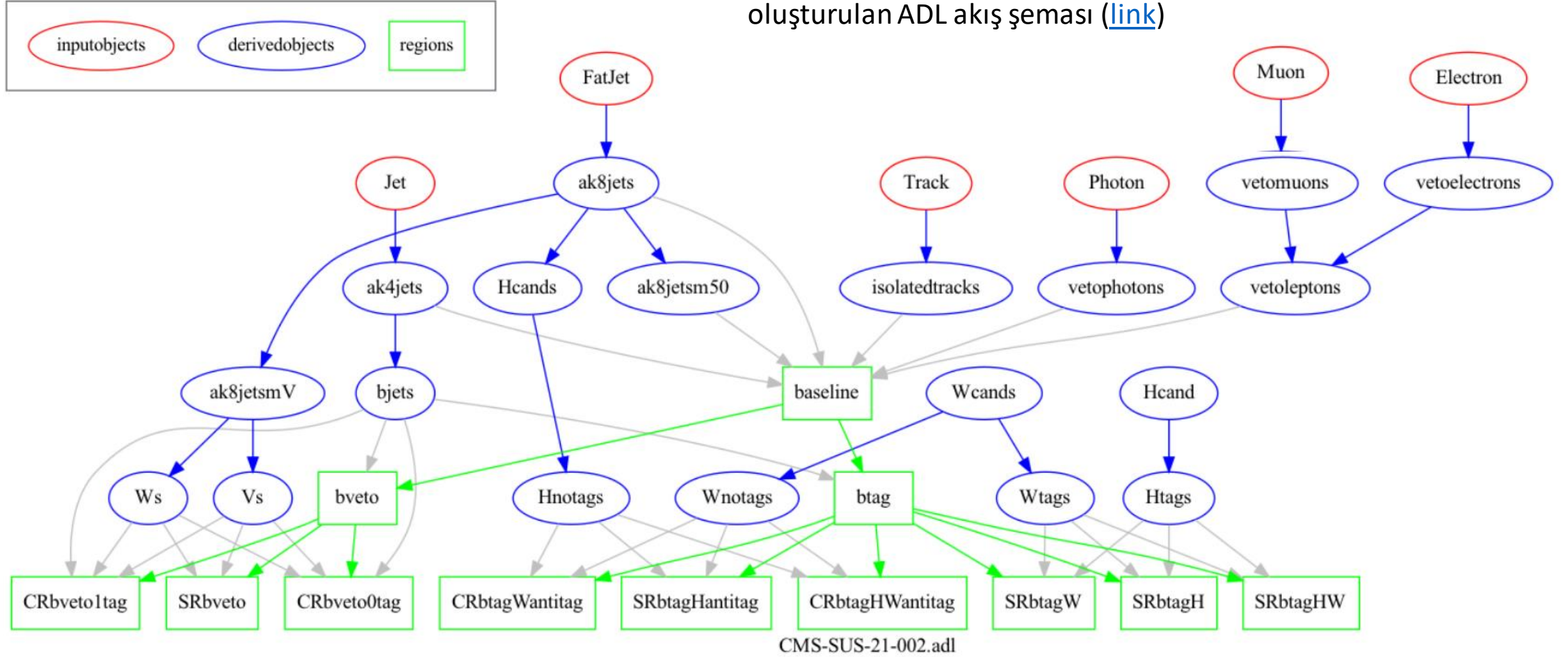
- Girdi verisindeki özellikler, harici fonksiyonlar gibi özellikler ADL sözdiziminden ayrıştırıldı:
  - Parçacıklar ve fonksiyonlar artık ADL ayrıştırıcısına(parser) sabit kodlanmak zorunda değil
  - CutLang harici fonksiyonları ve ntuple deęişkenlerini tanıyıp kullanabiliyor.
- ADL için **Soyut sözdizimi ağacı**(Abstract syntax tree) üretilebiliyor.
- Görsel araçlar:
  - Soyut sözdizimi ağacı kullanılarak analizler **akış şeması** haline getirilebiliyor.
  - **PyROOT** temelli **çizim araçları** geliştirildi
    - Farklı süreçler arasında **şekil kıyaslaması**
    - İki Süreç arasında **ağırlıklı kıyaslama**(örneğin sinyal ve ardalın)
    - Veri, farklı ardalınlar ve sinyallerin bir arada kıyaslandığı **ağırlıklı grafikler**(örneęi sonraki slaytlarda)
- Birçok **işlevsellik ve performans** geliřmeleri gerçekleştirildi.





## Yeni Özellik: Graphviz ile ADL akış şeması

Graphviz kullanılarak otomatik oluşturulan ADL akış şeması ([link](#))



## Yeni Özellik: Ntuple'daki değişkenler doğrudan okunabilir.

- Artık sabit bir veri tipine bağlı kalmadan, herhangi bir ROOT dosyasındaki tüm değişkenleri ADL dosyalarında **hızlı ve kolay bir şekilde** kullanmak mümkün.
  - Önceden tanımlamaya ihtiyaç yok
  - ADL sözdizimindeki her işlem sorunsuz bir şekilde yapılabilir(obje/değişken tanımları, olay seçimleri, histogramlama, matematik işlemleri...)

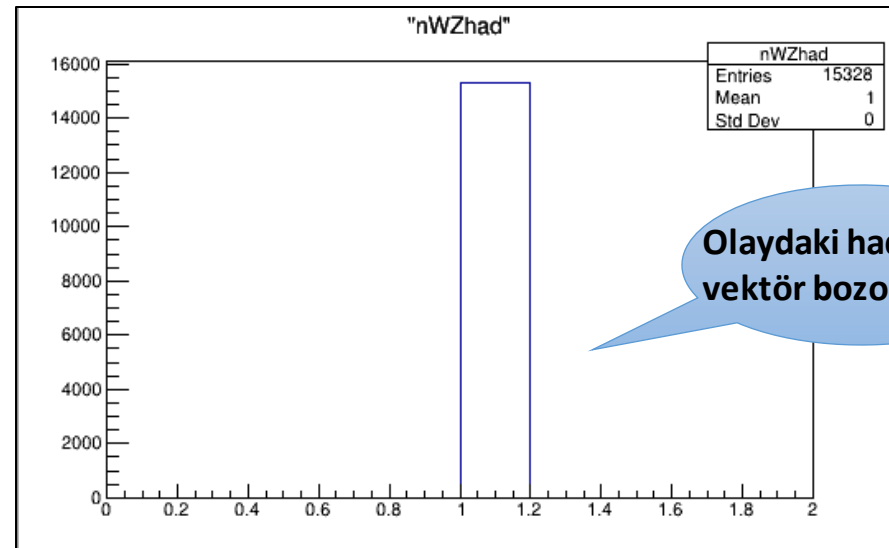
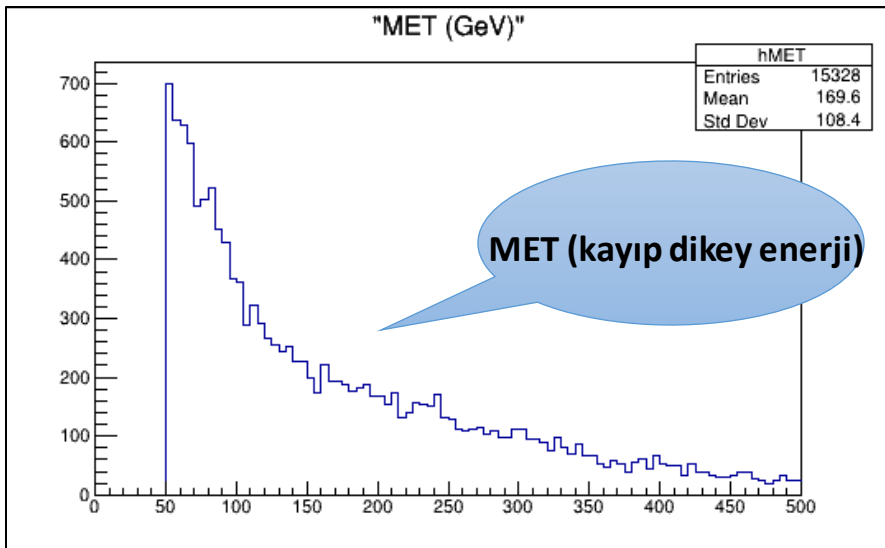
```
# OBJECTS
object goodJet : JET
select jets_pt(JET)*0.001 > 10
select jets_eta(JET) < 5.5

#EVENT SELECTION
region twojets
select ALL
weight xsec xs
select Size(goodJet) >= 2
select nTaus_OR == 0
select met_met*0.001 > 50
select nZmm == 0
select nWZhad == 1
histo hMET, "MET (GeV)", 100, 0, 500, met_met*0.001
histo nWZhad, "nWZhad", 10, 0, 2, nWZhad
```

Olay seçimleri

```
twojets Based on 200000 events:
ALL : 1 +- 0 evt: 200000
xsec xs : 0.01508 +- 0.000273 evt: 3016.584925
Size(goodJet) >= 2 : 0.8971 +- 0.00553 evt: 2706.314083
nTaus_OR == 0 : 0.9217 +- 0.00516 evt: 2494.534738
met_met*0.001 > 50 : 0.6873 +- 0.00928 evt: 1714.581623
nZmm == 0 : 0.9842 +- 0.00302 evt: 1687.417276
--> Overall efficiency = 0.844 % +- 0.0205 %
```

Sonuç çıktısı





## Yeniden Yorumlama(Reinterpretation)

**Yeniden yorumlama:** Yayınlanmış bir analizin, orijinal yayında yorumlanmamış fizik modellerini yorumlama amacıyla baştan yeniden kodlanmasıdır.

Yeniden yorumlama çalışmalarını genellikle fenomenoloji çalışanlar, deneysel sonuçların favori fizik modelleri üzerindeki etkisini görmek için yaparlar

- Bu çalışmalarda analiz kodu deneysel çerçevelere dahil değildir.
  - Halka açık simülasyon araçlarında dedektör bilgileri tüm detaylarıyla dahil olmadığı için yeniden kodlama orijinal deney kodu kadar isabetli olmaz.
- Sinyal modelleri için Monte Carlo simülasyon olayları üretilir.
- Tahmin edilen sinyal sayıları/verimlerini elde etmek için analiz kodu olaylar üzerinden çalıştırılır.
- Tahmin edilen sinyal sonucu, deney yayınındaki gözlenen veri ve aralan beklentisi sonuçları ile birlikte kullanılarak limit hesaplanır.



## ADL/CutLang ile YenidenYorumlama(Reinterpretation)

ADL, fenomenoloji topluluklarıyla bilgi alışverişini oldukça pratik hale getirir.

- Tüm analiz mantığını açık bir şekilde tarif eder.
- Deneylerden, doğrudan genel girdi tiplerine uyarlamayı mümkün kılar.
  - Deneydeki nesne tanımları, nümerik nesne ID'leri ile etiketleme(tagging) verimlerine dayalı basitleştirilmiş nesne tanımları ile değiştirilir.
  - Olay seçimleri neredeyse tamamen aynı kalır: Trigger seçimleri yerine trigger verimleri kullanılabilir.
- ADL dosyalarındaki analiz çıktılarını ifade etmek için genel sözdizimi mevcut:
- CutLang, var olan sonuçları analiz çıktılarıyla aynı formatta olan histogramlar şeklinde kaydeder.
  - Bu sayede cutflowlar kıyaslanabilir ve limit hesabı yapılabilir.
- ADL, HEPDATA veya benzeri platformlara analiz sonuçlarına ilişkin bilgilerin sağlanmasını kolaylaştırabilir.



## Doğrulama(Validation)

**Doğrulama:** Yeniden yorumlamanın doğru bir şekilde yapıldığının sağlamasıdır.

Yeniden yazılmış bir analiz, **orijinal analizin sonuçlarını** belirli bir dereceye kadar isabetle **yeniden üretebilir.**

Bir analizi doğrulamanın yolları:

- Eğer belirli bir sürece(sinyale) ait countlar veya cutflow mevcutsa:
  - Birkaç sinyal noktası için sinyal olaylarının üretilmesi.
  - Kendi count ve cutflowlarını elde etmek için bu olaylar üzerinde **kendi analiz kodunun çalıştırılması.**
  - Elde edilen count ve cutflowların deney tarafından yayınlanan sonuçlarla **kıyaslanması.**
- Eğer yalnızca limitler mevcutsa:
  - Birçok sinyal noktası için sinyal olaylarının üretilmesi.
  - **Analiz kodunun çalıştırılıp** count ve limitlerin hesaplanması.
  - Elde edilen limitlerin deney tarafından yayınlanan limitler ile **kıyaslanması.**



## ADL/CutLang ile Doğrulama(Validation) : Efficiency Map Creator

ADL/CutLang ile geniş ölçekli bir analiz uygulaması ve doğrulama çalışması başlattık. Ana odak noktası

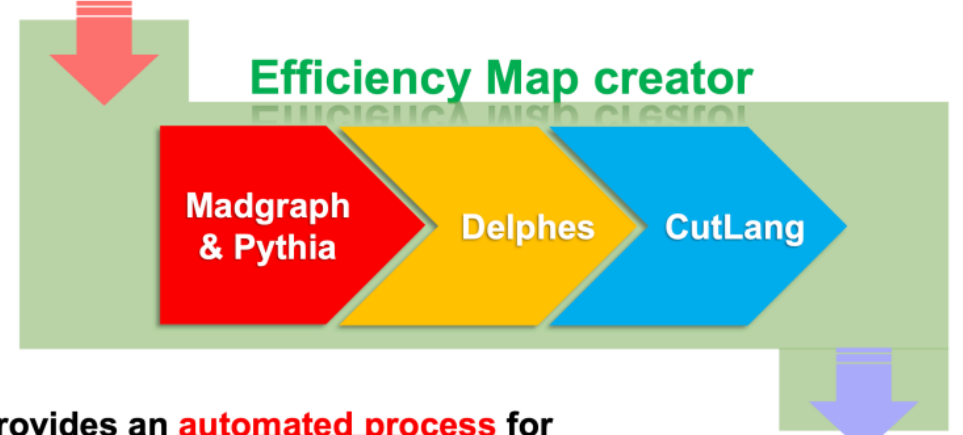
- Süpersimetri analizleri ancak Egzotik analizlerine de genişliyoruz.

Doğrulama çalışmaları için **SModelS Efficiency Map Creator**'u kullanıyoruz:

- Deneysel sonuçlarla kıyaslama yaparak analizleri doğrulamak için kullanılabilir.
- **Yapılandırılabilir kullanıcı arayüzü**: Hangi model ve kütle noktalarının üretileceği, hangi adımların çalıştırılacağı, hangi çıktının kaydedileceği belirtilebilir.
- EM-Creator, Wolfgang Waltenberger ve Jan Mrozek tarafından **ADL/CutLang ile birlikte çalışması için uyarlandı**.
  - Son adım: Verim haritaları
  - Limit hesaplama SModelS içerisinde gerçekleşiyor.

<https://smodels.github.io>

Number of events, ADL analysis file,  
mass range, SMS topology, ...



Provides an **automated process** for  
analysis through **simple command line**

Selected number of  
events / efficiencies in all  
regions & bins



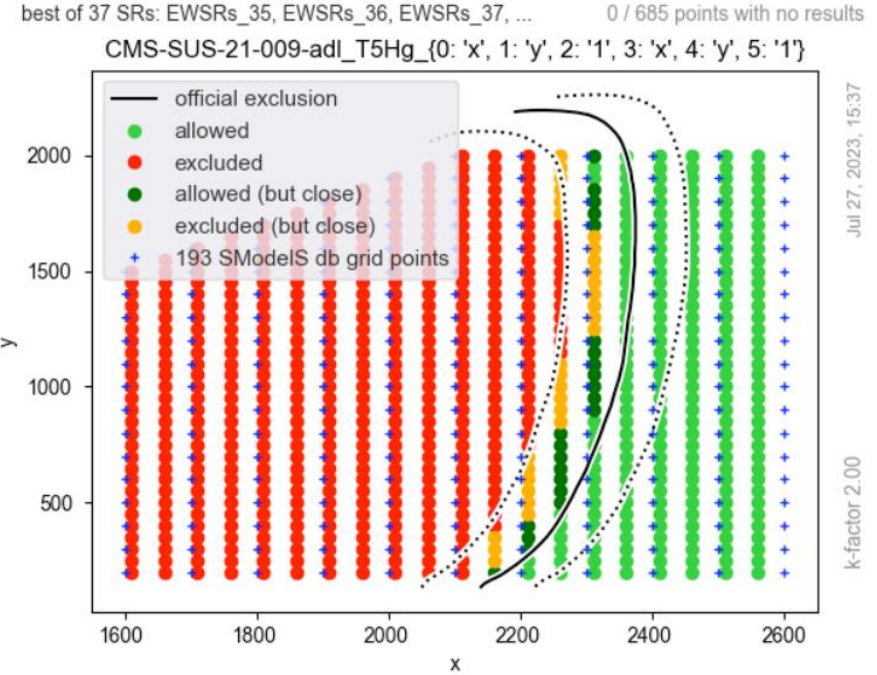
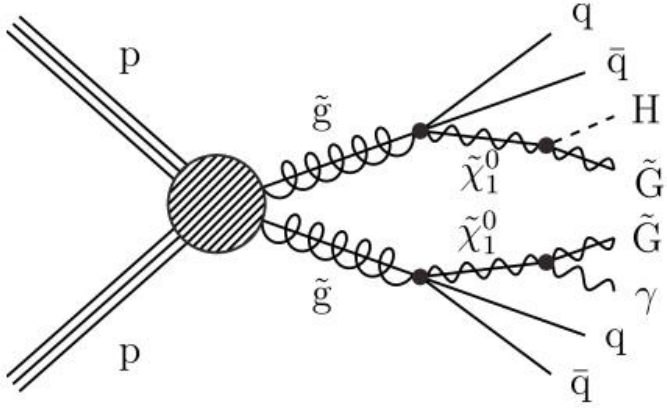


## ADL/CutLang ile Doğrulama(Validation) : Efficiency Map Creator

Son zamanlarda yayınlanan ATLAS ve CMS analizlerini doğrulama üzerine çalışıyoruz.

Güncel doğrulama örneği:

[CMS-SUS-21-009](#): “photons + multijets + MET”.



Devam eden doğrulama çalışmaları: [CMS-SUS-18-004](#), [CMS-SUS-21-006](#), [ATLAS-SUSY-2018-16](#), [ATLAS-SUSY-2019-09](#), [ATLAS-SUSY-2018-30](#), [ATLAS-SUSY-2020-27](#), [ATLAS-SUSY-2019-22](#), [ATLAS-SUSY-2018-22](#)





# Makine Öğrenimi Modellerinin Entegre Edilmesi

ADL/CutLang, ML modellerini çalıştırmak üzere adapte edildi.

- Şu an için .onnx modelleriyle çalışıyor. ( <https://onnx.ai/> )
- Onnx modellerini çalıştırmayı sağlayan ADL/CutLang sözdizimi:

```
# define the list of inputs
define listofinputs = {var1 var2 var3 ..... varN}
# define the ML output
define myMLvar = OME(my/directory/myfunc.onnx, listofinputs)
```

- Onnx fonksiyonlarının uygulandığı ATLAS analizleri:  
[ATLAS-SUSY-2019-04](#) (RPV leptons + jets) ve [ATLAS-SUSY-2018-30](#) (multi b + MET).



## ADL/CutLang ve LHC Open Data

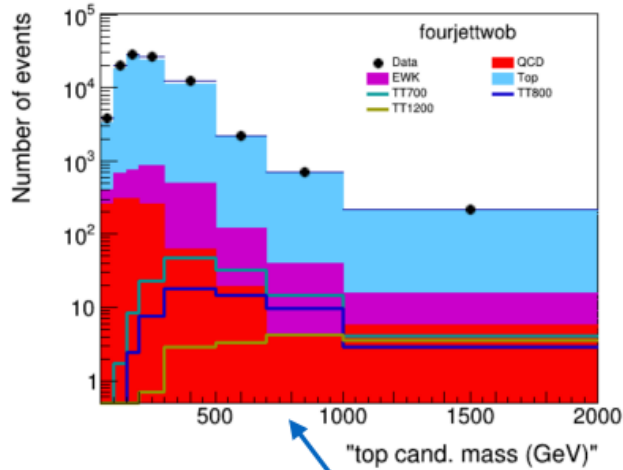
ADL/CutLang, [ATLAS ve CMS açık verileri](#) ile analizleri çalıştırmak için kullanılıyor.

- Yeniden yorumlama açısından kullanımı: [Yeni modellere olan hassasiyeti en yükseğe çıkartmak](#) için ADL veritabanındaki var olan analizleri optimize ediyoruz.
- Haziran 2023'te gerçekleşen CMS Open Data Workshop için ADL/CutLang kullanılarak [açık verileri kullanarak yeniden yorumlama'ya](#) yönelik ilk eğitim hazırlandı.
  - [Vektör benzeri T kuark sinyali için bir ttbar analizinin yeniden yorumlanması](#) üzerine çalışıldı.
  - Odak noktası, [VLT sinyaline olan duyarlılığı arttırmak](#) için analizin [yeniden optimize](#) edilmesi.  
[Eğitimin linki](#)
  - İlgili açık veri ve MC olayları üzerinde çalışıyor.
  - [CutLang, ROOT, açık veriler ve VNC'yi](#) barındıran bir docker konteynerinde çalışıyor
- [2022 CMS Open Data Workshop](#)'taki eğitimde [CMS'in Run 1 verisiyle gerçekleştirdiği 'Boosted W ve Higgs bozonlarıyla vektör-benzeri kuark analizi'](#) yeniden uyarlanmıştır.
  - [Eğitimin linki](#)

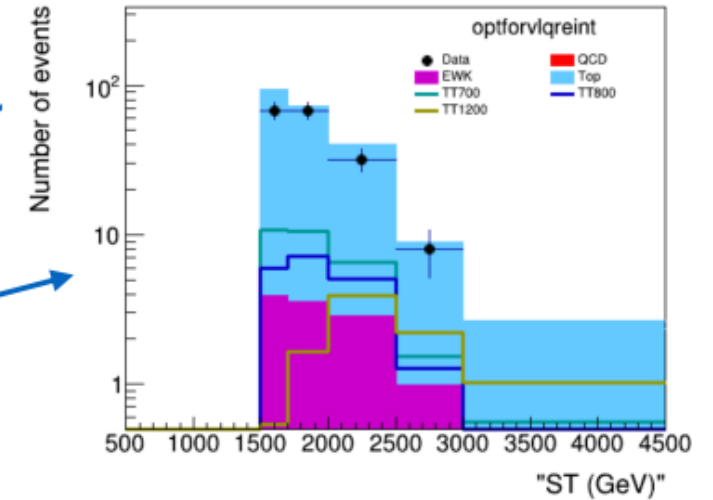
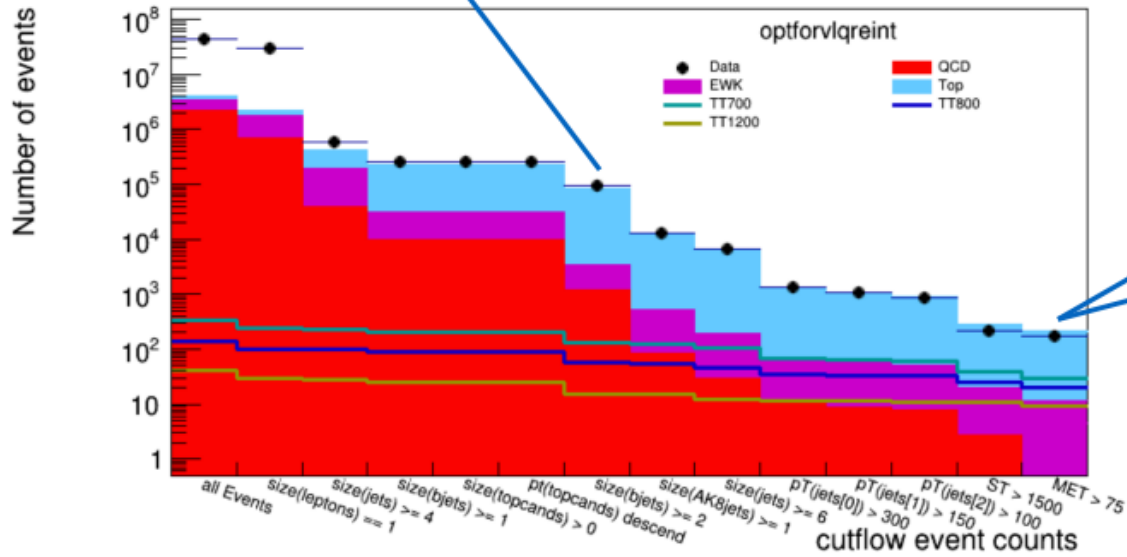
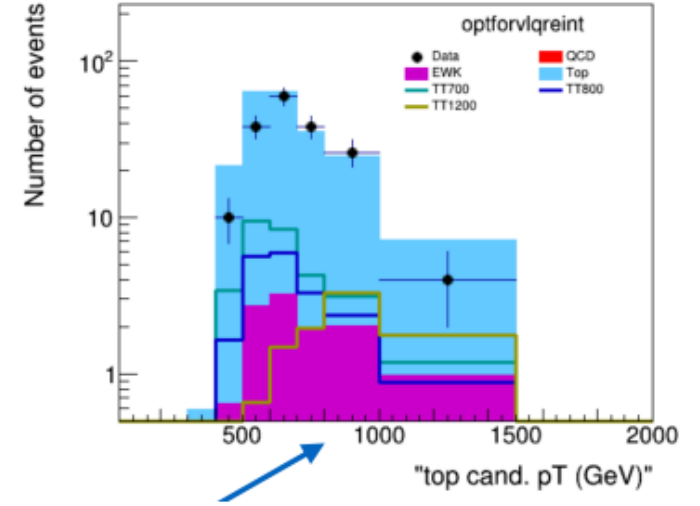


# ADL/CutLang ve LHC Open Data

Data, BG and 2 VLT signals vs. top candidate mass for the t $\bar{t}$  analysis selection.



Top candidate mass  $p_T$  and ST after reoptimizing the t $\bar{t}$  selection by adding several new cuts. (B2G-16-024 uses ST)

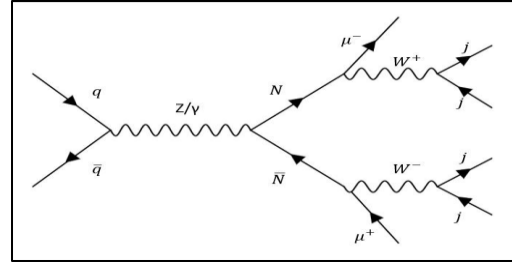


Cutflow histogramları CutLang tarafından otomatik olarak üretiliyor.

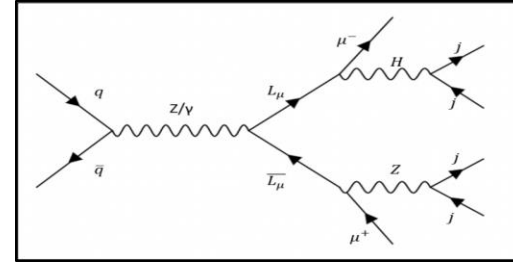


# ADL/CutLang ile Vektör-benzeri ağır lepton analizi

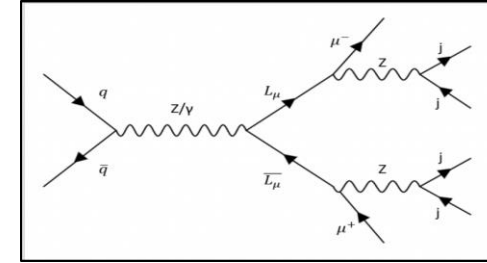
Büyük Hadron Çarpıştırıcısı tarafından 13 TeV kütle merkezi enerjisinde 139 fb<sup>-1</sup> toplam ışınılığa karşılık gelen, ATLAS algıcının 2015-2018 yılları arasında kaydettiği veride ADL/CutLang kullanarak E6 Büyük Birleşme Kuramı'nın öngördüğü vektör benzeri ağır leptonlar (VLL) araştırılmaktadır.



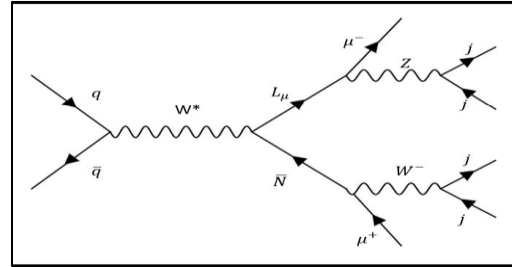
$pp \rightarrow N\bar{N} \rightarrow W^+ \mu^- W^- \mu^+ \rightarrow 2\mu 4j$



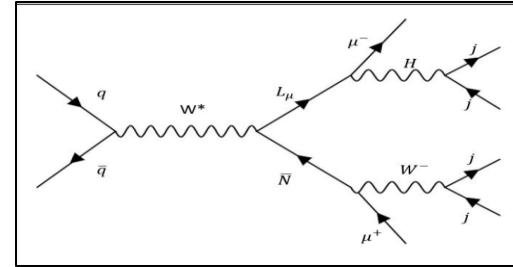
$pp \rightarrow L\bar{L} \rightarrow H\mu^- Z\mu^+ \rightarrow 2\mu 4j$



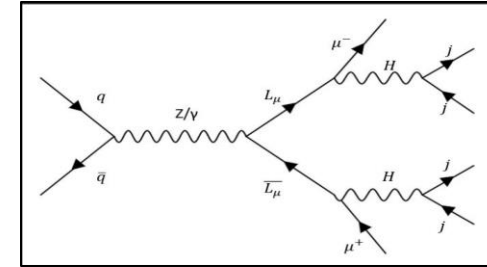
$pp \rightarrow L\bar{L} \rightarrow Z\mu^- Z\mu^+ \rightarrow 2\mu 4j$



$pp \rightarrow L\bar{N} \rightarrow Z\mu^- W^- \mu^+ \rightarrow 2\mu 4j$



$pp \rightarrow L\bar{N} \rightarrow H\mu^- W^- \mu^+ \rightarrow 2\mu 4j$



$pp \rightarrow L\bar{L} \rightarrow H\mu^- H\mu^+ \rightarrow 2\mu 4j$

- Son durumda zıt yüklü 2 lepton içeren sinyal bölgesine ait seçimler:

	OS leptons	Pt(leps) > 20 GeV	Tight ID and FC Looselsolation for electrons	Medium ID and FC Looselsolation for muons	Tau veto	Z veto	1Z	nJets >= 2	Mll > 15 GeV
SR	✓	✓	✓	✓	✓	✓	✗	✓	✓



## ADL/CutLang ile Vektör-benzeri ağır lepton analizi

- Son durumda zıt yüklü 2 lepton içeren bozunum kanallarında hadronik bozonlar ve ağır leptonlar,  $M_L = 500 \text{ GeV}$  kütleli sinyal Monte Carlo örneği kullanılarak yeniden yapılandırılmıştır. En iyi adayları bulmak için  $\chi^2$  algoritması kullanıldı.

$$\chi^2 = \frac{(m_{LH}^{reco} - m_{LZ}^{reco})^2}{(\sigma_{LH})^2 + (\sigma_{LZ})^2} + \frac{(m_H^{reco} - m_H^{real})^2}{(\sigma_H)^2} + \frac{(m_Z^{reco} - m_Z^{real})^2}{(\sigma_Z)^2}$$

### #EVENT VARIABLE DEFINITIONS

```
define jetA : goodJet[-1]
define jetB : goodJet[-2]
define jetC : goodJet[-3]
define jetD : goodJet[-4]
define muA : goodMuo[-1]
define muB : goodMuo[-2]
```

Herhangi ikişer jetten oluşan Higgs ve Z bozonlarının tanımlanması

```
define Zhad2 : jetC jetD
define Hhad1 : jetA jetB
```

Ağır leptonların tanımlanması

```
define Lz2 : Zhad2 muB
define Lh1 : Hhad1 muA
```

```
define HZchi : (m(Hhad1) - 125)^2 / 10^2 + (m(Zhad2) - 91)^2 / 8^2
define totchiHZ : (m(Lh1) - m(Lz2))^2 / (27.4^2 + 27.5^2) + HZchi
```

$\chi^2$  algoritması

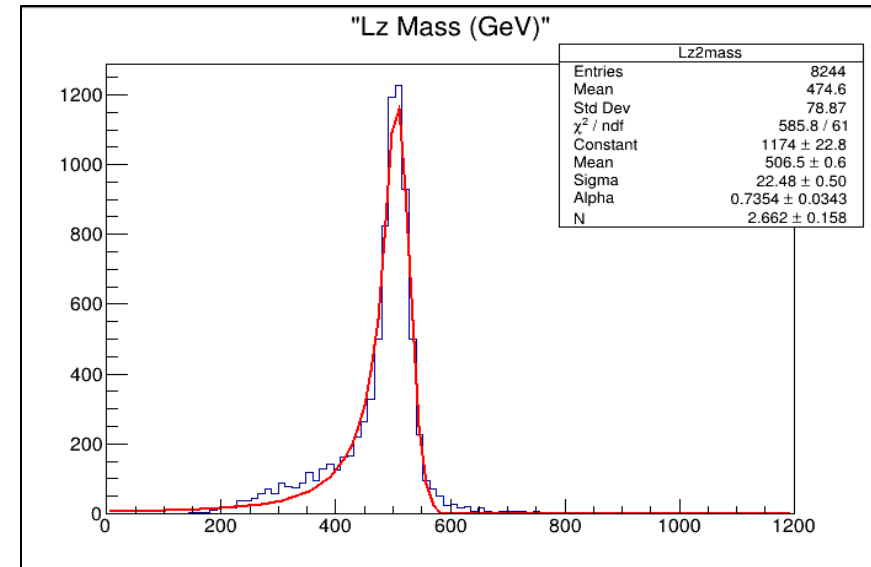
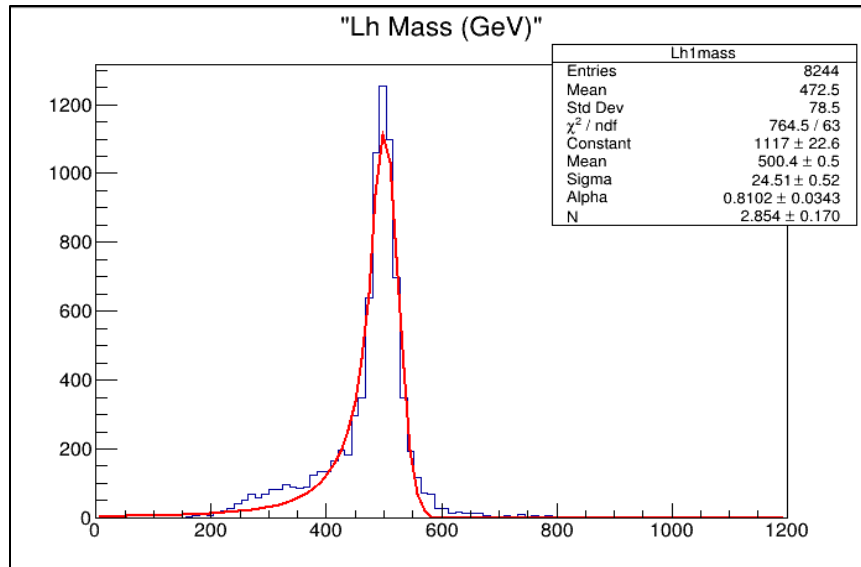
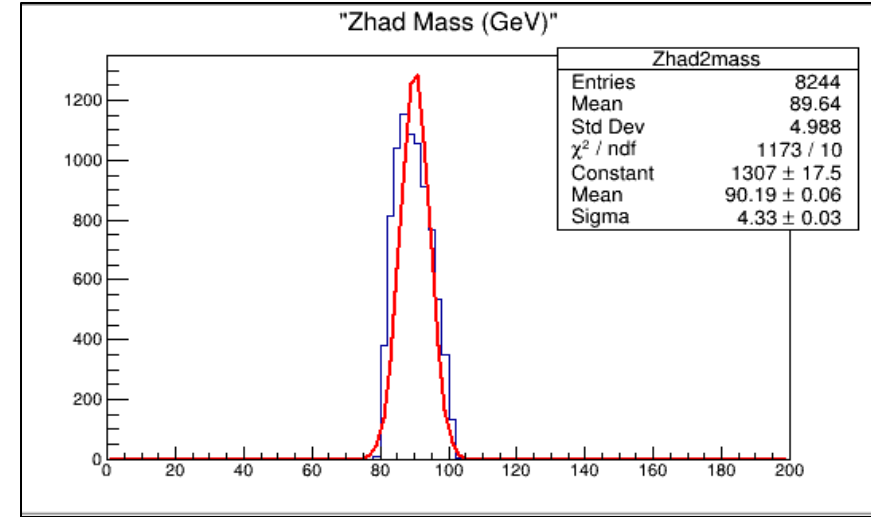
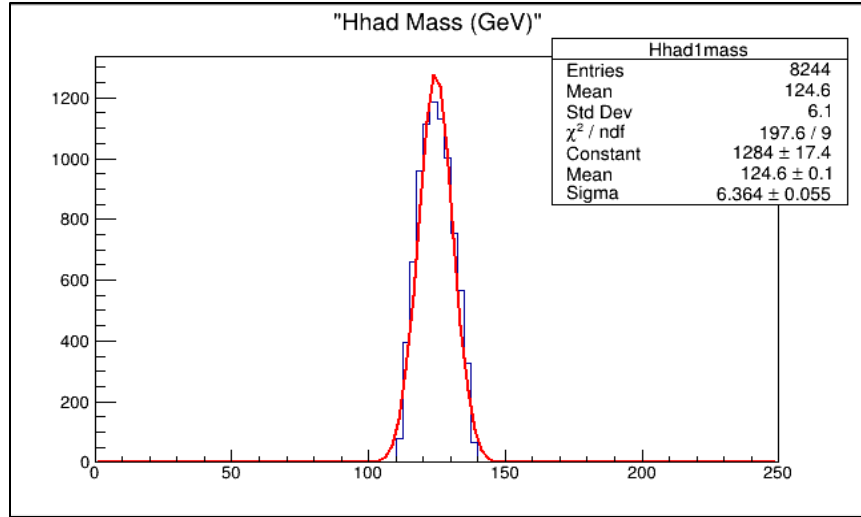
```
region HZ2L0S
preselection
select totchiHZ <= 0
histo Hhad1mass, "Hhad Mass (GeV)", 100, 0, 250, m(Hhad1)
histo Zhad2mass, "Zhad Mass (GeV)", 100, 0, 200, m(Zhad2)
histo Lh1mass, "Lh Mass (GeV)", 100, 0, 1200, m(Lh1)
histo Lz2mass, "Lz Mass (GeV)", 100, 0, 1200, m(Lz2)
```

$\chi^2$  minimizasyonu



# ADL/CutLang ile Vektör-benzeri ağır lepton analizi

- Yeniden yapılandırılan Higgs, Z bozonlarının ve ağır leptonların kütle dağılımları ( $M_L = 500 \text{ GeV}$ )





# GPT4 DEMO

```
#CLA TwoJetTwoMuonEvents
```

```
object goodMuon : MUON  
select Pt(MUON) > 15  
select AbsEta(MUON) < 2.4
```

```
object goodJet : JET  
select Pt(JET) > 30  
select AbsEta(JET) < 2.4
```

```
define Zcandidate : goodMuon[0] goodMuon[1]  
define Wcandidate : goodJet[0] goodJet[1]  
define chargedHiggsCandidate : Zcandidate Wcandidate
```

```
region TwoJetTwoMuonAnalysis  
select ALL  
select Size(MUON) >= 2  
select Size(goodMuon) == 2  
select Size(JET) >= 2  
select Size(goodJet) == 2  
histo hZcandidateMass, "Invariant mass of Z candidate (GeV)", 100, 60, 120, m(Zcandidate)  
histo hWcandidateMass, "Invariant mass of W candidate (GeV)", 100, 40, 140, m(Wcandidate)  
histo hChargedHiggsCandidateMass, "Invariant mass of charged Higgs candidate (GeV)", 100, 100, 500, m(chargedHiggsCandidate)
```

GO

Select events with two jets, two muons. Make a Z from the muons and make a W from two jets. Plot the invariant masses of both bosons. Then add the two bosons as a charged higgs candidate. Plot the invariant mass of the new object.



Here is the updated CutLang analysis that selects events with two jets and two muons, creates a Z boson candidate from the muons, a W boson candidate from the jets, and finally, a charged Higgs candidate from the Z and W bosons. The invariant masses of the Z and W boson candidates and the charged Higgs candidate are plotted in histograms:





# GPT4 DEMO

#CLA OneTauOneMuonEvents

```
object goodMuon : MUON
select Pt(MUON) > 15
select AbsEta(MUON) < 2.4
```

```
object goodTau : TAU
select Pt(TAU) > 20
select AbsEta(TAU) < 2.3
```

```
object goodJet : JET
select Pt(JET) > 30
select AbsEta(JET) < 2.4
```

```
object goodBJet : JET
select Pt(JET) > 30
select AbsEta(JET) < 2.4
select JETbtag > 0.89
```

```
algo OneTauOneMuonSelection
select ALL
select Size(MUON) == 1
select Size(goodMuon) == 1
select Size(TAU) == 1
select Size(goodTau) == 1
select Size(JET) == 3
select Size(goodJet) == 3
select Size(goodBJet) >= 1
```



Select events with one tau and one muon. then select 3 jets and a b jet.



Here's a CutLang analysis that selects events with exactly one tau, one muon, three jets, and at least one b jet:

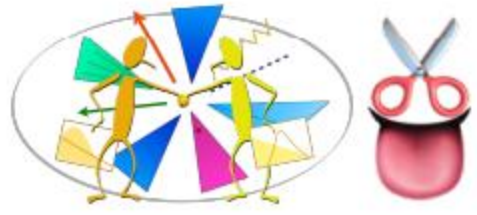


But I wanted to have exactly 1 bjet



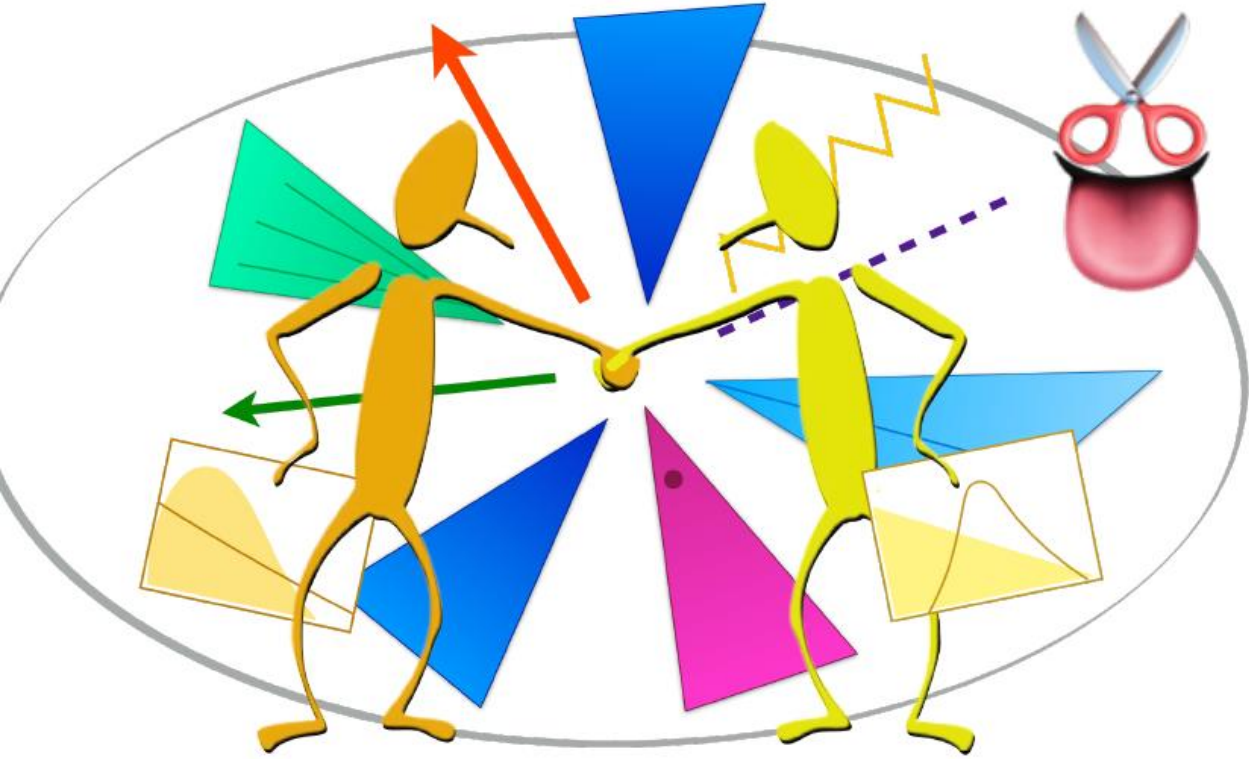
My apologies for the confusion. The analysis can be modified to select events with exactly one b jet as follows:

```
algo OneTauOneMuonSelection
select ALL
select Size(MUON) == 1
select Size(goodMuon) == 1
select Size(TAU) == 1
select Size(goodTau) == 1
select Size(JET) == 3
select Size(goodJet) == 3
select Size(goodBJet) == 1
```



## Sonuç olarak

- ADL/CutLang, çok amaçlı ve pratik bir analiz yaklaşımı sunuyor.
- ADL/CutLang, yeniden yorumlama(reinterpretation) çalışmaları için oldukça uygundur.
- Hali hazırda büyük ölçekte analiz yeniden-yorumlama/doğrulama çalışmaları sürmektedir.
- SUSY analizlerini doğrulamak için SModelS EM-Creator çerçevesini kullanıyoruz.
- PyROOT (ve jupyter) tabanlı çizim(plotting) araçları mevcuttur.
- Biri optimizasyon yoluyla yeniden yorumlamayı içeren 2 adet tam ölçekli CMS Open Data eğitimi mevcuttur.
- Aktif olarak 2 ATLAS analizi (vektör-benzeri kuarklar ve vektör-benzeri leptonlar) yürütülmektedir.
- ADL sözdizimi ve derleyicisindeki altyapı çalışmaları hızla devam ediyor.
  - ML modelleri eklendi.
  - ADL anahtar kelimeleri ve ntuple değişkenleri birbirlerinden ayrıştırıldı
  - Ve daha birçok stabilite ve işlevsel gelişmeler devam ediyor...



## İlave Slaytlar

[cern.ch/adl](http://cern.ch/adl)



## ADL Sözdizimi: ana bloklar, anahtar kelimeler, işlemciler

Blokun amacı	Anahtar kelime
Nesne tanımlama blokları	<b>object</b>
Olay seçme blokları	<b>region</b>
Analiz veya ADL bilgisi	<b>info</b>
Tablo	<b>table</b>

Anahtar kelime amacı	Anahtar kelime
değişkenleri ve sabitleri tanımlama	<b>define</b>
nesne veya olay seçimi	<b>select</b>
nesne veya olay reddetme	<b>reject</b>
ana nesneyi tanımlama	<b>take</b>
ağırlıkları uygulama	<b>weight</b>
değişkenleri yazdırma	<b>print</b>
nesneleri sıralama	<b>sort</b>
histogramları tanımlama	<b>histo</b>
değişkenleri kaydetme	<b>save</b>

İşlem	İşlemci
Karşılaştırma işlemcileri	<b>&gt; &lt; =&gt; =&lt; == != [] (dahil) [] (hariç)</b>
Matematiksel işlemciler	<b>+ - * / ^</b>
Mantıksal işlemciler	<b>and or not</b>
Koşul işlemcisi	<b>koşul ? doğru : yanlış</b>
Optimizasyon işlemcileri	<b>~= (en yakın) ~! (en uzak)</b>
Lorentz vektörleri toplama	<b>LV1 + LV2 LV1 LV2</b>

ADL sözdizimi kuralları ve kullanım örnekleri:

<https://twiki.cern.ch/twiki/bin/view/LHCPhysics/ADL>

# Histogram ve Çizim Araçları



ADL/CutLang geniş histogramlama kabiliyetlerine sahiptir:

- 1D ve 2D sabit bin ve değişken binli histogramlar tek satırda tanımlanır.
- Farklı regionlar ve kesim seviyelerinde tekrar tekrar kullanılacak histogram listeleri tanımlanabilir.
- Cutflow histogramları ve analiz bin histogramları otomatik olarak çiziliyor.

Histogram Listesi

```
# HISTOGRAM LISTS
histoList jetHistos
histo hnjets , "number of jets", 20, 0, 20, size(goodJets)
histo hjet1pt , "jet1 pT (GeV)", 40, 0, 1000, pT(goodJets[0])
histo hjet2pt , "jet2 pT (GeV)", 40, 0, 1000, pT(goodJets[1])
histo hjet3pt , "jet3 pT (GeV)", 40, 0, 1000, pT(goodJets[2])
histo hjet1eta , "jet1 eta", 40, -3, 3, eta(goodJets[0])
histo hjet2eta , "jet2 eta", 40, -3, 3, eta(goodJets[1])
histo hjet3eta , "jet3 eta", 40, -3, 3, eta(goodJets[2])
```

```
# EVENT SELECTION
region baseline
select ALL
select size(goodJets) >= 3
select size(goodbJets) >= 1
select pT(goodJets[0]) > 200
# Fixed bin 1D:
histo hmet, "MET (GeV)", 40, 0, 1000, MET
# Fixed bin 2D:
histo hj1ptMET, "MET vs. jet1 pT (GeV)", 40, 0, 1000, 40, 0, 1000, pT(goodJets[0]), MET
# Variable bin 1D (no , between bin edges!):
histo hmetvarbin, "MET (GeV) variable bin", 0.0 10.0 20.0 50.0 100.0 500.0, MET
# Use histolists after different levels of selection:
select MET > 100
jetHistos
select MET > 200
jetHistos
```



# Counts and Cutflows



- Counts fonksiyonuyla **deneye ait cutflow değerleri** ADL dosyasına dahil edilebilir.
- CutLang bu sayıları kullanarak deney sonuçları için bir cutflow histogramı üretir.

```
# preselection region
region presel
select ALL
counts sigone 100.0 +- 0.8 , 100.0 +- 0.5 , 100.0 +- 0.0 , 100.0 +- 0.5
counts sigtwo 100.0 +- 0.0 , 100.0 +- 0.1 , 100.0 +- 0.1 , 100.0 +- 0.1
counts sigthree 100.0 +- 0.2 , 100.0 +- 0.5 , 100.0 +- 0.5
counts sigfour 100.0 +- 0.0 , 100.0 +- 0.1 , 100.0 +- 0.2
select size(jets) >= 2
counts sigone 100.0 +- 0.8 , 100.0 +- 0.5 , 100.0 +- 0.0 , 100.0 +- 0.5
counts sigtwo 100.0 +- 0.0 , 99.3 +- 0.1 , 99.6 +- 0.1 , 100.0 +- 0.1
counts sigthree 99.9 +- 0.2 , 98.8 +- 0.5 , 99.1 +- 0.5
counts sigfour 99.6 +- 0.0 , 95.4 +- 0.1 , 97.8 +- 0.2
select HT > 300
counts sigone 100.0 +- 0.8 , 100.0 +- 0.5 , 100.0 +- 0.0 , 100.0 +- 0.5
counts sigtwo 90.1 +- 0.4 , 74.8 +- 0.5 , 82.0 +- 0.3 , 94.6 +- 0.4
counts sigthree 98.7 +- 0.4 , 98.3 +- 0.5 , 98.9 +- 0.6
counts sigfour 72.2 +- 0.3 , 58.2 +- 0.3 , 83.0 +- 0.4
select MHT > 300
counts sigone 85.5 +- 2.7 , 86.8 +- 1.9 , 77.1 +- 0.5 , 83.0 +- 2.1
counts sigtwo 13.8 +- 0.4 , 19.9 +- 0.5 , 21.2 +- 0.4 , 22.2 +- 0.7
counts sigthree 74.5 +- 1.2 , 79.6 +- 1.4 , 88.1 +- 1.4
counts sigfour 9.2 +- 0.2 , 13.6 +- 0.2 , 31.3 +- 0.5

countsformat sigone
process T1tttt1900200, "T1tttt 1900 200", stat
process T1bbbb1800200, "T1bbbb 1800 200", stat
process T1qqqq1300100, "T1qqqq 1300 100", stat
process T5qqqqVV1800100, "T5qqqqVV 1800 100", stat

countsformat sigtwo
process T1tttt13001000, "T1tttt 1300 1000", stat
process T1bbbb13001100, "T1bbbb 1300 1100", stat
process T1qqqq12001000, "T1qqqq 1200 1000", stat
process T5qqqqVV14001100, "T5qqqqVV 1400 1100".

countsformat bgests
process lostlep, "Lost lepton background", stat, syst
process zinv, "Z --> vv background", stat, syst
process qcd, "QCD background", stat, syst

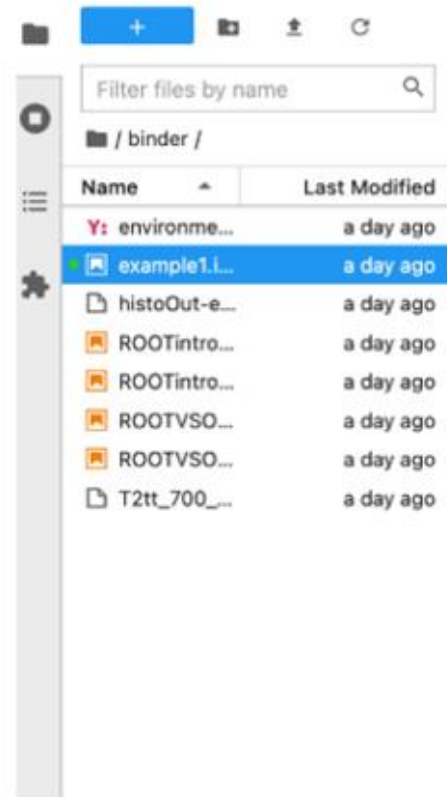
countsformat results
process est, "Total estimated BG", stat, syst
process obs, "Observed data"

region searchbins
presel
# Table 3, 1-10
bin MHT [] 300 350 and HT [] 300 600 and size(jets) [] 2 3 and size(bjets) == 0
counts bgests 38870 +- 320 +- 580 , 89100 +- 200 +- 2600 , 1800 +- 1000 + 1200 - 800
counts results 129800 +- 1100 +- 2800 , 130718
bin MHT [] 300 350 and HT [] 600 1200 and size(jets) [] 2 3 and size(bjets) == 0
counts bgests 2760 +- 61 +- 39 , 4970 +- 50 +- 150 , 330 +- 180 +- 160
counts results 8060 +- 200 +- 220 , 7820
bin MHT [] 300 350 and HT >= 1200 and size(jets) [] 2 3 and size(bjets) == 0
counts bgests 181 +- 17 +- 3 , 308 +- 12 +- 18 , 62 +- 34 +- 27
```



# CutLang Birçok Ortamda Çalışıyor

	Linux	MacOS	Windows
Native	✓	✓	
Docker	✓	✓	✓
Conda cmd line	✓	✓	
Conda -> Jupyter	✓	✓	
Web -> Jupyter	✓	✓	✓



```

[ ]: !cd ../runs
[ ]: !wget https://www.dropbox.com/s/zza28peyjj8qgg6/T2tt_700_50.root

[ ]: %cutlang file=T2tt_700_50.root filetype=DELPHES name=exHistos
# ADL file for ROOT101 tutorial: opening files and drawing histograms
# Runs with binder/ROOTIntroPython.ipynb, binder/ROOTIntroCpp.ipynb

# Object selection
object jets
take jet
select pT(jet) > 30
select abs(eta(jet)) < 2.4

# Event selection
region presel
select ALL
select MET > 300
select size(jets) >= 4
histo hjet1pt, "jet 1 pT (GeV)", 40, 0, 1000, pT(jets[0])
histo hjet2pt, "jet 2 pT (GeV)", 40, 0, 1000, pT(jets[1])
histo hjet3pt, "jet 3 pT (GeV)", 40, 0, 1000, pT(jets[2])
histo hjet4pt, "jet 4 pT (GeV)", 40, 0, 1000, pT(jets[3])
histo hmetjet1pt, "MET vs. jet 1 pT", 20, 300, 1300, 20, 0, 1000, MET, pT(jets[0])
  
```





# CutLang Birçok Ortamda Çalışıyor



- Jupyter/Binder arayüzüyle CutLang' i mobil cihazlarda çalıştırmak da mümkün!



```
08:53
hub.gke2.mybinder.org
ROOT example1 Visit repo Copy Bind
Menu Not Trusted ROOT C++ with CutLang
Code Memory: 528.8 MB / 2 GB
Download GitHub Binder
In [ ]:
!cd ../runs
!wget https://www.dropbox.com/s/zza28pe
In [ ]:
%%cutlang file=T2tt_700_50.root filetype
# ADL file for ROOT101 tutorial: openin
# Runs with binder/ROOTIntroPython.ipyn
# Object selection
object jets
take jet
select pT(jet) > 30
select abs(eta(jet)) < 2.4
# Event selection
region presel
select ALL
select MET > 300
select size(jets) >= 4
histo hjet1pt, "jet 1 pT (GeV)", 40,
histo hjet2pt, "jet 2 pT (GeV)", 40,
histo hjet3pt, "jet 3 pT (GeV)", 40,
```

```
10:52
hub.gke2.mybinder.org
ROOT example1 Visit repo Copy Bind
Menu Not Trusted ROOT C++ with CutLang
Code Memory: 539.7 MB / 2 GB
Download GitHub Binder
object jets
take jet
select pT(jet) > 30
select abs(eta(jet)) < 2.4
# Event selection
region presel
select ALL
select MET > 300
select size(jets) >= 4
histo hjet1pt, "jet 1 pT (GeV)", 40,
histo hjet2pt, "jet 2 pT (GeV)", 40,
histo hjet3pt, "jet 3 pT (GeV)", 40,
histo hjet4pt, "jet 4 pT (GeV)", 40,
histo hmetjet1pt, "MET vs. jet 1 pT",
Forced number of entries 2
0000
starting entry 0
Processing event 0
Processing event 5000
Processing event 10000
Processing event 15000
Efficiencies for analysis
: BP_1
presel Based on 20000 eve
```

```
10:54
hub.gke2.mybinder.org
ROOT example1 Visit repo Copy Bind
Menu Trusted ROOT C++ with CutLang
Code Memory: 539.8 MB / 2 GB
Download GitHub Binder
# Event selection
region presel
select ALL
select MET > 300
select size(jets) >= 4
histo hjet1pt, "jet 1 pT (GeV)", 40,
histo hjet2pt, "jet 2 pT (GeV)", 40,
histo hjet3pt, "jet 3 pT (GeV)", 40,
histo hjet4pt, "jet 4 pT (GeV)", 40,
histo hmetjet1pt, "MET vs. jet 1 pT",
ALL : 1 +- 0
evt: 20000
MET > 300 : 0.5647 +- 0.
00351 evt: 11295
size(jets) >= 4 : 0.811 +-
- 0.00368 evt: 9160
[Histo] jet 1 pT (GeV) :
1 +- 0 evt: 91
60
[Histo] jet 2 pT (GeV) :
1 +- 0 evt: 91
60
```



## Basit nesneleri yeniden yazmak

Deneysel analizler, var olan tanımlarda basitleştirme yapılarak yeniden kullanılabilir.

### b-tagging for UL NanoAODv9

```
# AK4 jets
object JetAK4
take Jet
select jetID(Jet) > 0
select pt(Jet) > 30
select abs(eta(Jet)) < 2.4
```

```
# b-tagged jets - medium
```

```
object MediumBTag
take JetAK4
select btagDeepB(JetAK4) >= 0.2598
```



### b-tagging for public use, e.g. with Delphes

```
# AK4 jets
object JetAK4
take Jet
# select jetID > 0
select pt(Jet) > 30
select abs(eta(Jet)) < 2.4
```

```
# b-tagged jets - medium
```

```
object MediumBTag
take JetAK4
select applyhitandmiss( btagdeepBmediumeff( pt, abs(eta) ) ) == 1
```

Verimleri, nesne özelliklerini ve hit&miss metodu uygulayan genel bir fonksiyon

CMS tarafından sağlanan verimler. ADL tablo halindeki verimleri kullanabilir.

Ayrıntılı bilgi ilave slaytlarda



## ADL Sözdizimi: fonksiyonlar

**Standart/dahili fonksiyonlar:** Yeterince genel matematik ve YEF işlemleri dilin ve onu yorumlayan herhangi bir aracın parçası olabilir.

- Matematik fonksiyonları: `abs()`, `sqrt()`, `sin()`, `cos()`, `tan()`, `log()`, ...
- Collection reducers: `size()`, `sum()`, `min()`, `max()`, `any()`, `all()`, ...
- YEF spesifik fonksiyonlar: `dR()`, `dphi()`, `deta()`, `m()`, ...
- Nesne ve koleksiyon işleme: `union()`, `comb()`, ...

**Harici/kullanıcı fonksiyonları:** Mevcut işlemcileri kullanarak ifade edilemeyen değişkenler ve standart fonksiyonlar, bir ADL dosyasından çağrılabilen bağımsız fonksiyonlarla ifade edilebilir ve bir veri tabanı aracılığıyla derleyiciler tarafından erişilebilir.

- Basit olmayan algoritmalara sahip değişkenler: `MT2`, `aplanarity`, `razor` değişkenleri, ...
- Analitik olmayan değişkenler: nesne ve tetikleme verimlilikleri, değişken/ML ile hesaplanan değişkenler ve verimlilikler, ...



## Nesne Verimleri

- Nesnelerin nitelikleri ve bunların belirsizliğine karşılık gelen verimlilik değerleri, tablo aracılığıyla ADL dosyasına kaydedilebilir.
- CutLang bu verimlilikleri analiz nesnelere 'hit-and-miss' yöntemi ile uygular.

```
object bjets
  take jets
  select abs(flavor(jets)) == 5
  select applyHM( btagdeepCSVmedium( Pt(jets) ) == 1)
```

```
table btagdeepCSVmedium
tabletype efficiency
nvars 1
errors true
# val      err-      err+      pTmin      pTmax
0.5790     0.0016     0.0016     -10.4      30.0
0.6314     0.0013     0.0013     30.0       35.0
0.6442     0.0011     0.0011     35.0       40.0
0.6596     0.0007     0.0007     40.0       50.0
0.6727     0.0007     0.0007     50.0       60.0
0.6812     0.0008     0.0008     60.0       70.0
0.6855     0.0008     0.0008     70.0       80.0
0.6873     0.0009     0.0009     80.0       90.0
0.6881     0.0010     0.0010     90.0       100.0
0.6880     0.0008     0.0008     100.0      125.0
0.6867     0.0011     0.0011     125.0      150.0
0.6826     0.0015     0.0015     150.0      175.0
0.6734     0.0020     0.0020     175.0      200.0
0.6624     0.0026     0.0026     200.0      225.0
0.6494     0.0034     0.0034     225.0      250.0
0.6419     0.0044     0.0044     250.0      275.0
0.6301     0.0054     0.0054     275.0      300.0
0.6202     0.0051     0.0051     300.0      350.0
```