GPUs for Higgs boson data analysis at the LHC using the Matrix Element Method

GPU in HIGH ENERGY PHYSICS
PISA
10-12 SEP 2014

Doug Schouten (TRIUMF)
Adam DeAbreu (SFU)
Bernd Stelzer (SFU)

Measuring Small Signals at Hadron Colliders

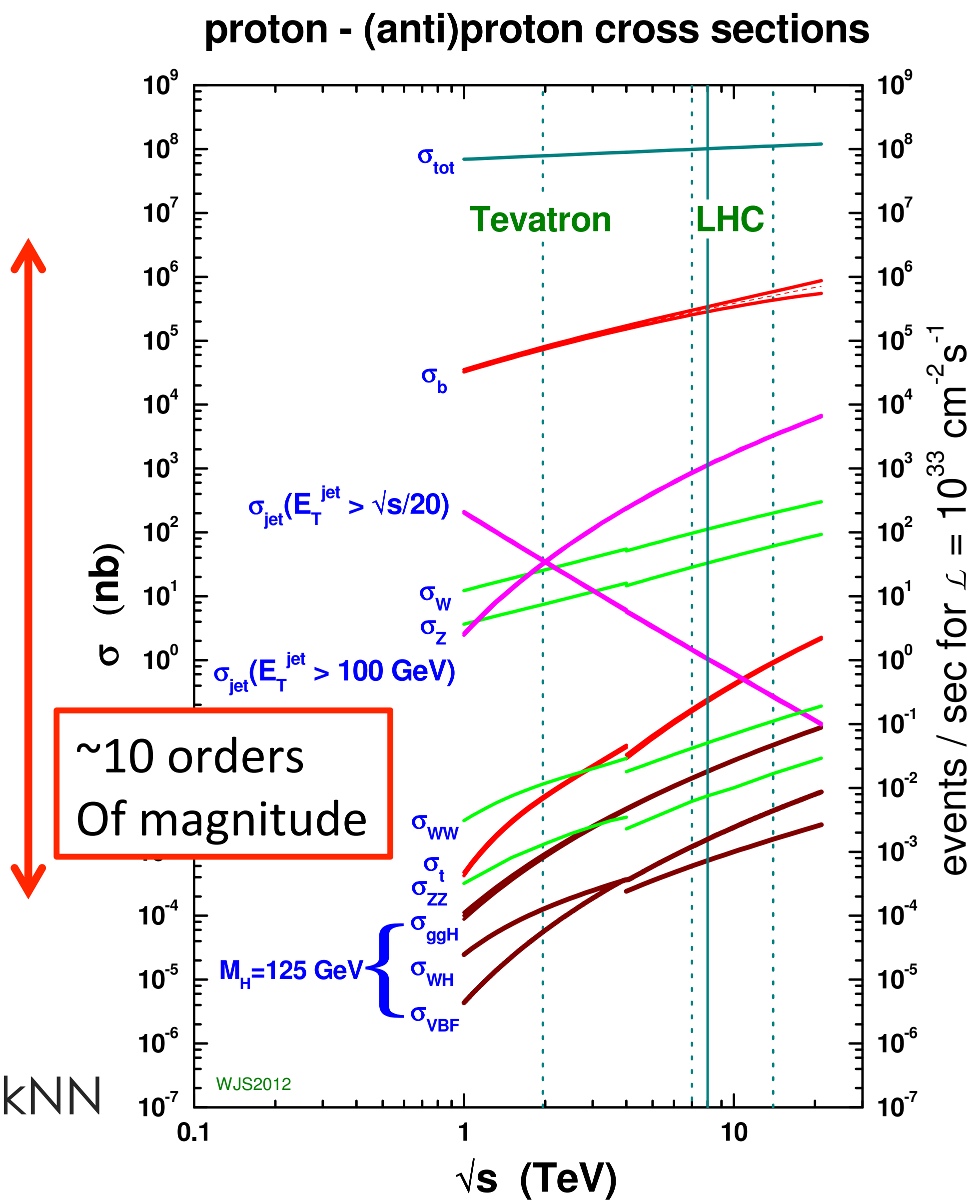
- Many *interesting processes* at hadron colliders *are rare*! (*Higgs, weak bosons, top quarks*)
- Before doing anything, challenging Signal : Background (S:B) ~ 1:10^{10}
- *First step:*

  Trigger and ID clean objects (e.g. leptons) → Improves S:B by ~ 10^6
- *Second step:*

  Characterize the data in the full dimensionality of the final state to *discriminate small signals from much larger backgrounds*

  ✓Topological event selection cuts

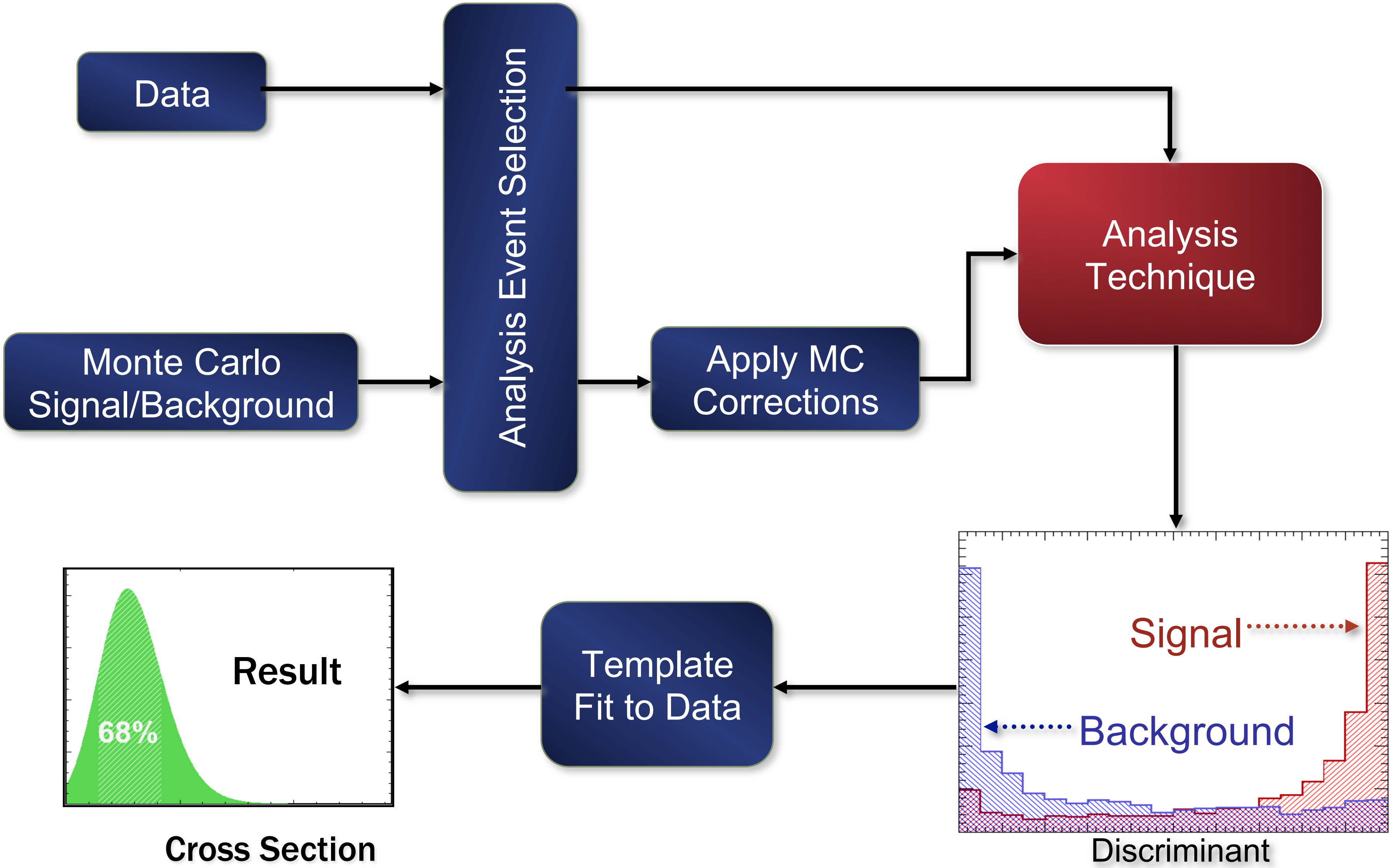
  ✓ Advanced analysis techniques



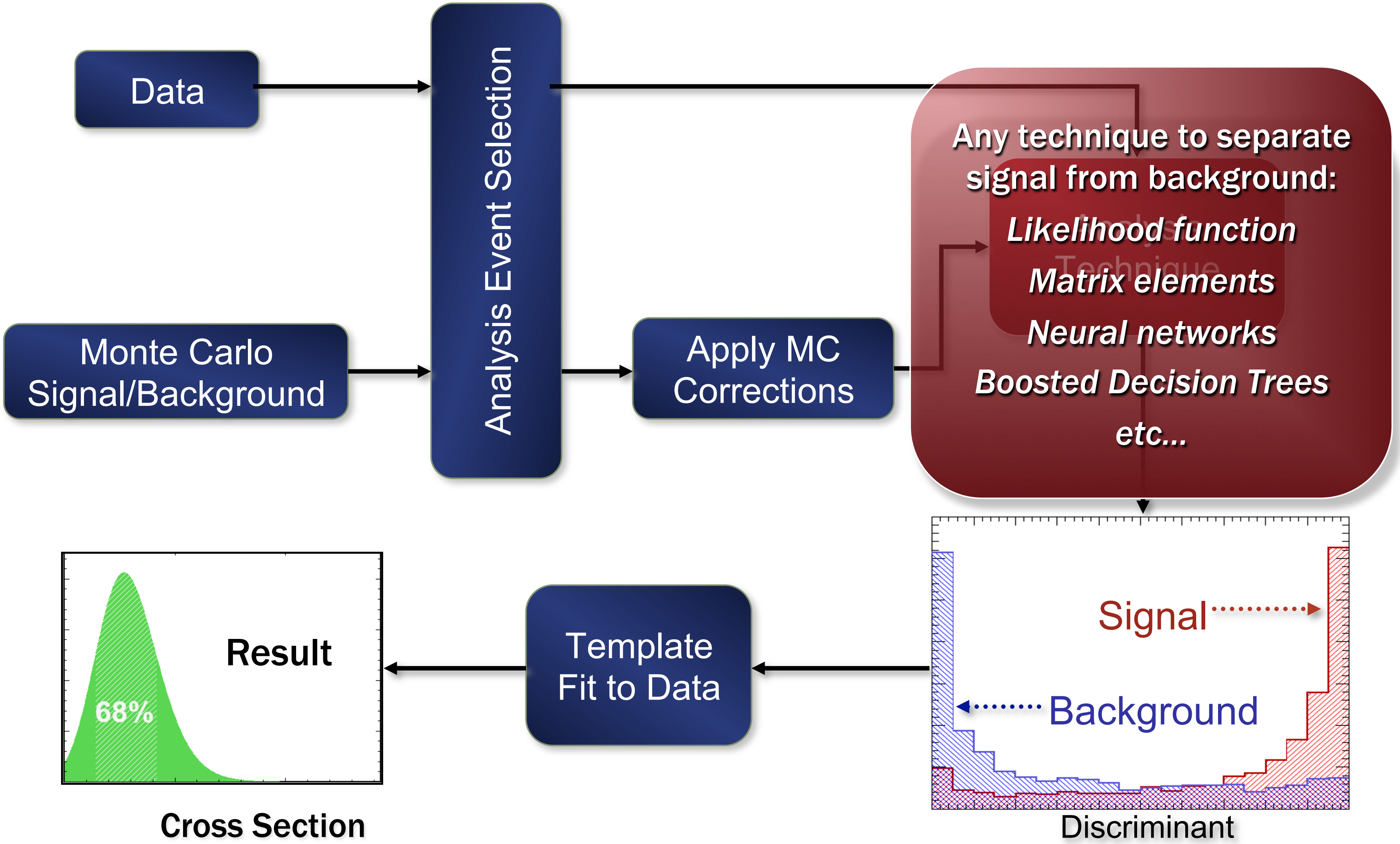
Machine learning: NN, BDT, SVM, kNN

Physics: Matrix Element Method

Simplified Multivariate Analysis Workflow

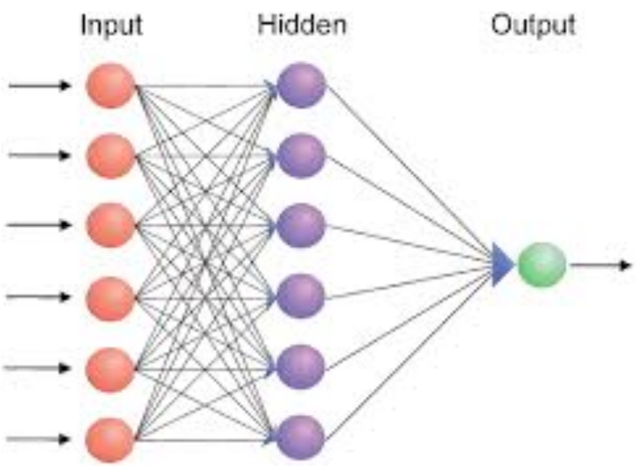


Simplified Multivariate Analysis Workflow



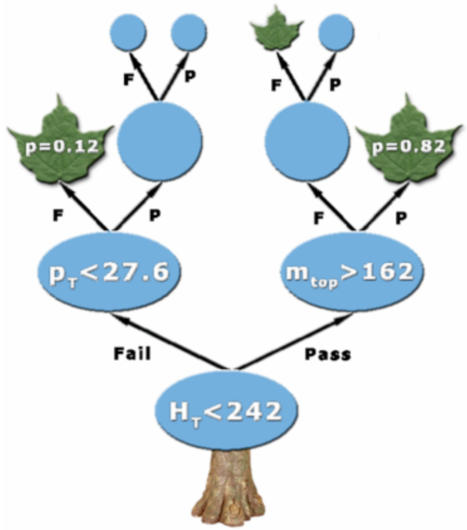
Machine Learning Algorithms used in HEP

Neural Networks



System of interconnected "neurons" computing values from input features trained by minimizing error function.

Boosted Decision Trees



Training of a sequence of cuts (hyper-cubes of phase space) to maximize the purity of output leafs.

Matrix Element Method

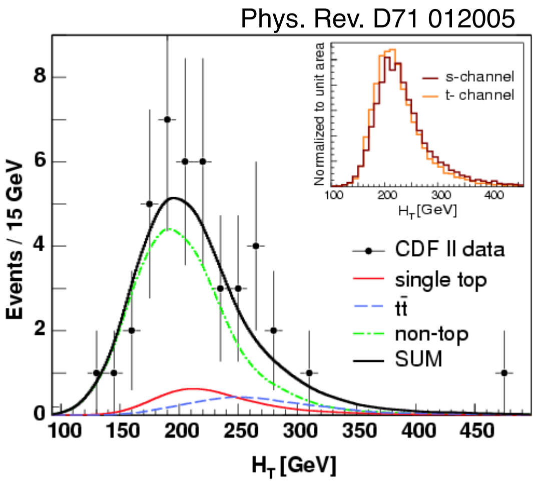
$$\boxed{P(x) = \frac{1}{\sigma}\left|M(p_i^{\mu})\right|^2 d\Phi}$$

Compute signal and background probability densities for each candidate event based on Fermi's Golden rule.

Machine learning algorithms require *training* from simulated data

The method 'knows all the physics'

Enabling Discoveries – Example Single Top

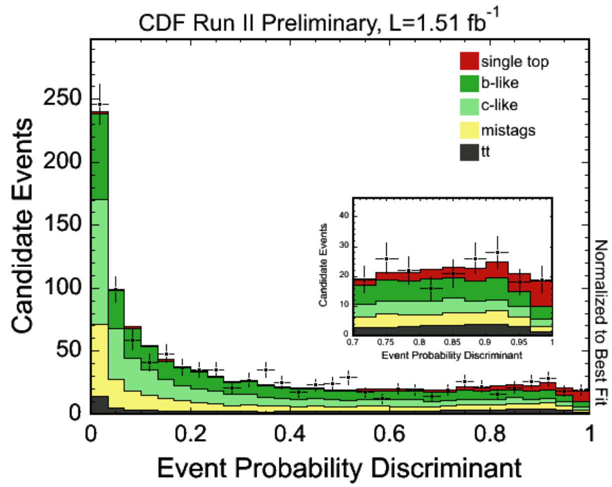


First Tevatron Run II result using 162 pb^{-1}

$\sigma_{\text{single top}}$ **< 17.5 pb at 95 % C.L.**

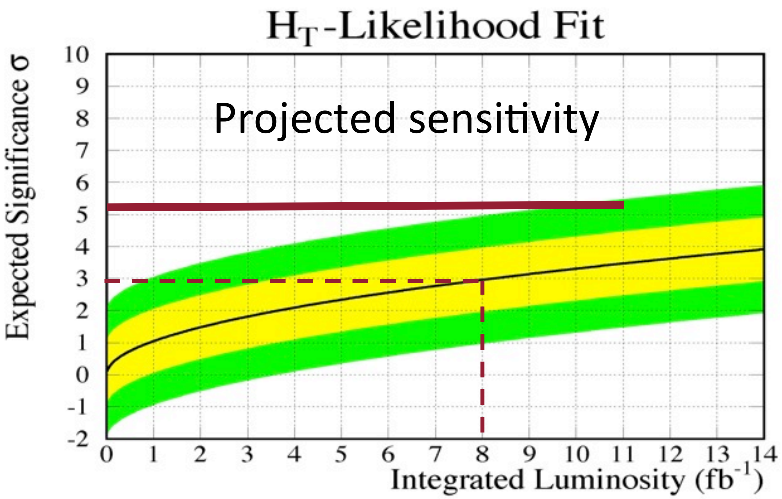
+ TIME

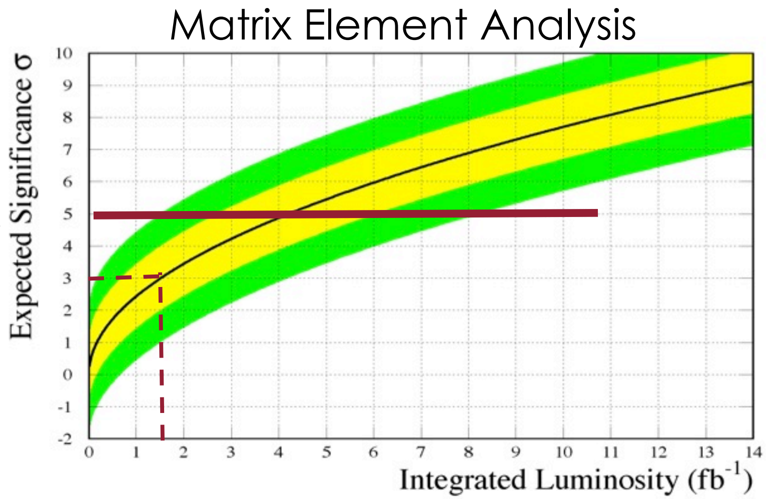
- *Development of powerful analysis techniques (Matrix Element, Neural Networks, Decision Trees)*
- *Continuous b-tagging to purify signal region*



2007: Evidence for single top quark production using 1.5 fb^{-1} (expected and observed!)

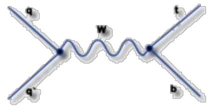
2009: Discovery using 3.2 fb^{-1}

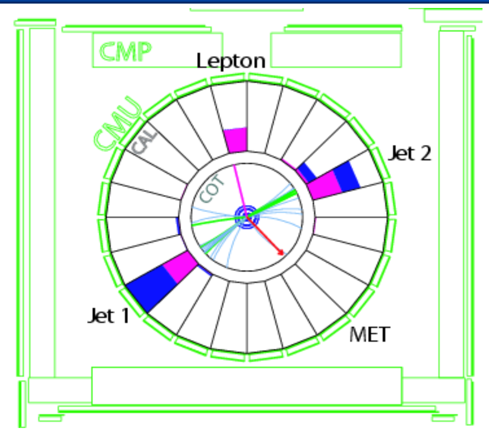




Single top would not have been discovered at the Tevatron w/o advanced analysis techniques

Matrix Element Method

- Attempt to encode all available kinematic / dynamic information about an event into a single observable
- Based on Fermi's Golden rule: P = |  |^2 · Phase space



$$P_i = \frac{1}{\sigma_{obs}} \sum_{flavour} \int_{V_n} M_i^2(\mathbf{Y}) \frac{f_1(x_1, Q^2) f_2(x_2, Q^2)}{|\vec{q}_1| \cdot |\vec{q}_2|} d\Phi_n(q_1 + q_2; y_1, ..., y_n) \cdot TF(\mathbf{Y};X)$$

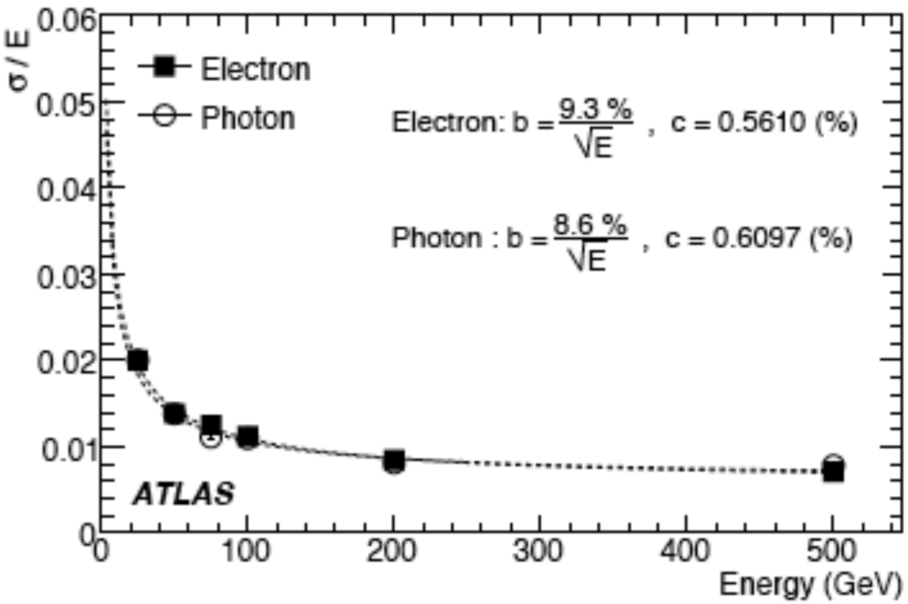
M_i	Lorentz invariant ME	$\mathbf{Y}$	Momenta of initial and final state particles	TF	Transfer functions
$f_1, \ f_2$	PDFs of colliding partons	$x_1, \ x_2$	Fractions of proton beam energy		Probability of measuring the set of observables (X) that correspond to particle kinematics (Y)
$q_1, \ q_2$	Momenta of colliding partons	σ_{obs}	Total cross-section for observed final state		

Phase space term

$$d\Phi_n(q_1 + q_2; y_1, .., y_n) = (2\pi)^4 \delta^4(q_1 + q_2 - \sum_{i=1}^{n} y_i) \prod_{i=1}^{n} \frac{d^3 y_i}{(2\pi)^3 2E_i}$$

Transfer Functions

- The transfer functions describes the evolution from *particle* → *observable*
- I.e. the transfer function TF(Y; X), provides the probability of measuring the set of observable variables (X) that correspond to the set of production variables (Y).
  - ✧X = measured momenta
  - ✧Y = particle momenta

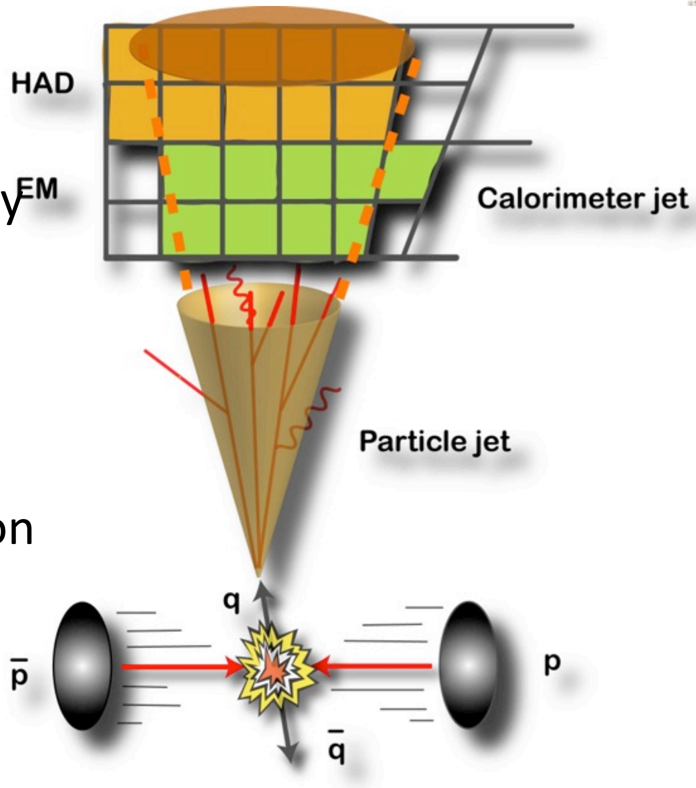


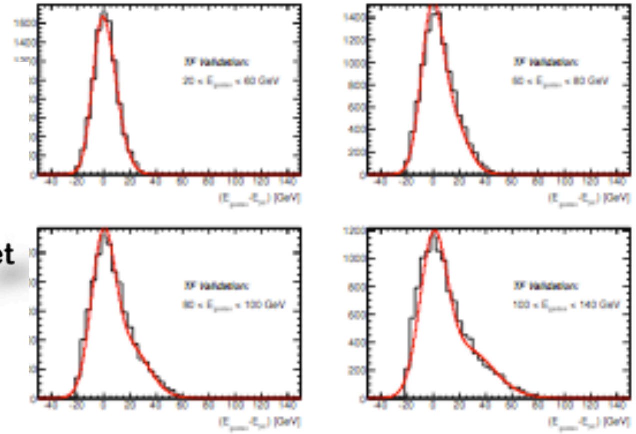
- Leptons are well measured.
  Jet directions are well measured
- The typical choice is to assume delta functions for lepton momenta and jet directions and only treat hadronic jet energies
- Flat TF for unobserved particles (e.g. neutrinos)

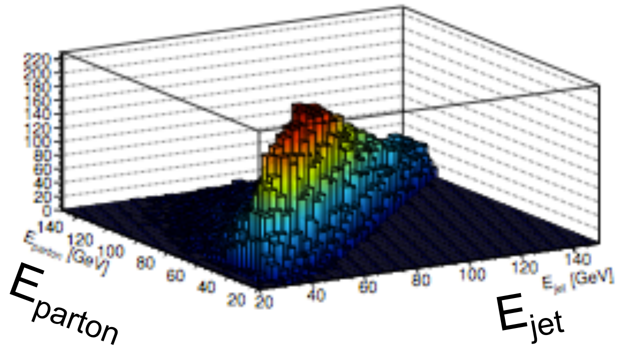
$$TF(Y;X) = \delta^3(\vec{p_l}^{\,y} - \vec{p_l}^{\,x}) \prod_{i=1}^{2} \delta^2(\Omega_i^y - \Omega_i^x) \prod_{j=1}^{2} TF\ (E_{parton_j}, E_{jet_j})$$

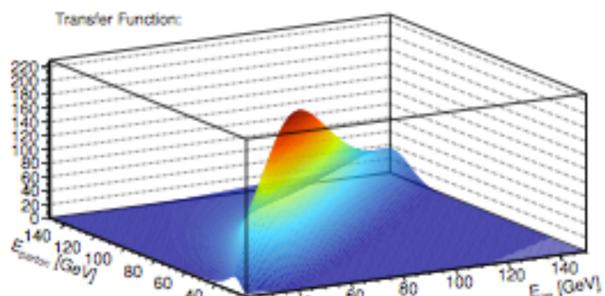
Transfer Functions for Hadronic Jets

- Calorimeters measure jet energies
- Relate particle jet energy to parton energy
- Double Gaussian parameterization , accounts for detector response (Gaussian core) and also for parton fragmentation outside of the jet definition (non-Gaussian tail).







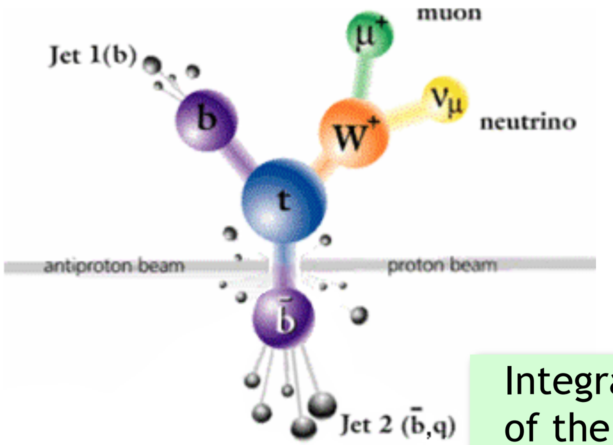


Double Gaussian parameterization:

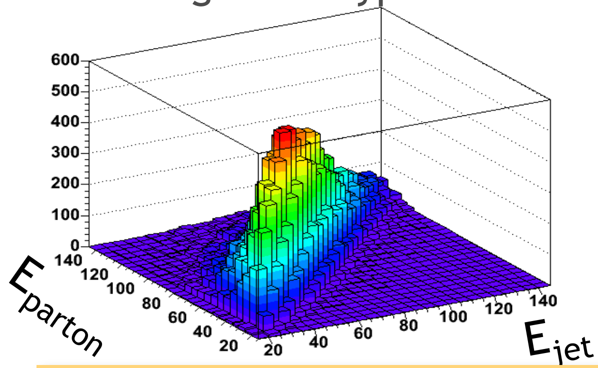
$$TF_{jet}(E_{jet}, E_{parton}) = \frac{1}{\sqrt{2\pi}(p_1 + p_2 p_5)}[\exp\frac{-(\delta_E - p_1)^2}{2p_2^2} + p_3 \exp\frac{-(\delta_E - p_4)^2}{2p_5^2}$$

where: $p_i = a_i + b_i E_{parton}$

Matrix Element Method



Event probability for signal and background hypothesis:



Try both jet-parton assignments and sum

Matrix Element

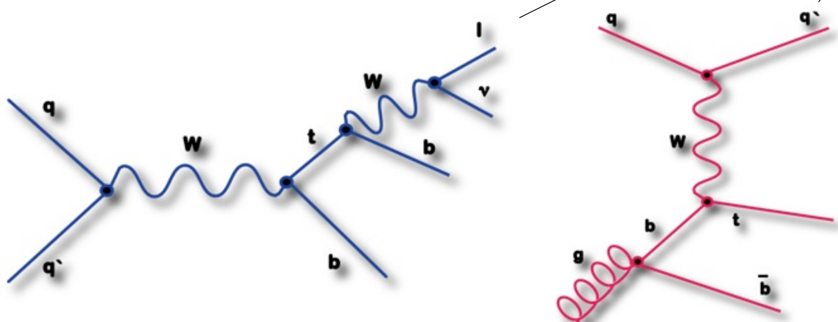
Integration over part of the phase space Φ

Jet Parton Transfer Function

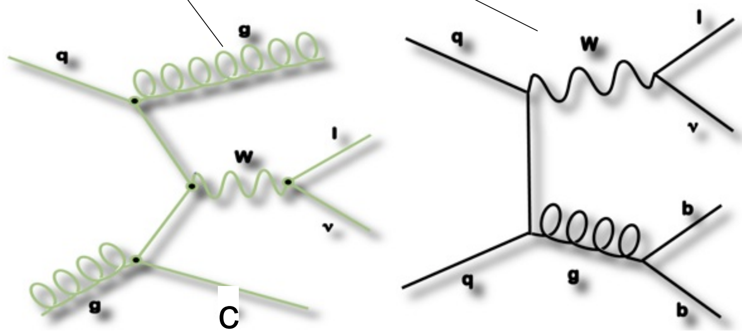
$$P(p_l^\mu, p_{jet1}^\mu, p_{jet2}^\mu) = \frac{1}{\sigma}\overbrace{\int dE_{part1}\, dE_{part2}\, dp_\nu^z} \sum_{comb} |M(p_i^\mu)|^2 \, d\phi \, \frac{f(q_1) f(q_2)}{|q_1||q_2|} TF(E_{jet}, E_{part})$$

Input 4-vectors of Lepton and 2/3 jets

Parton distribution functions



Signal

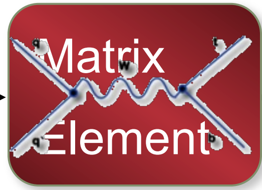


Background

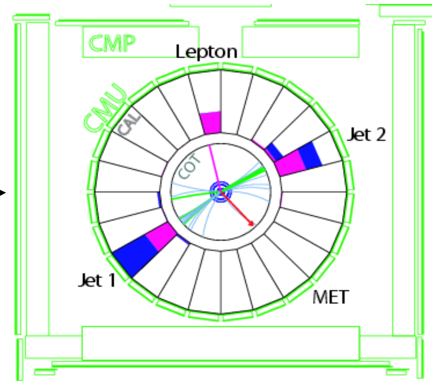
Matrix Element Method

Standard Event Simulation (Monte Carlo):

Random Phase Space Point dΦ →

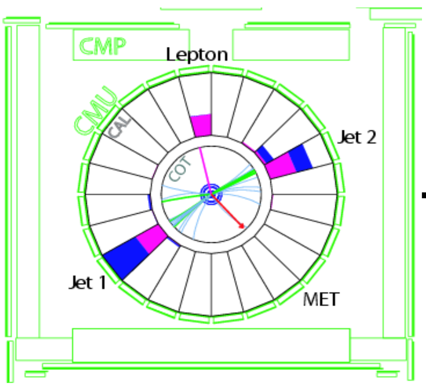


→ Detector Simulation →

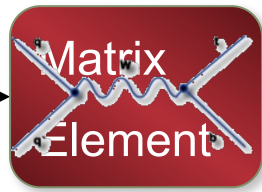


Produces events with process specific Kinematics/dynamics

Matrix Element Analysis:



→ Transfer Functions →



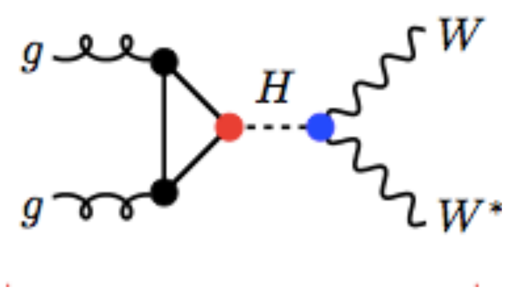
→ Event Probability

Transfer functions relates *observables to particle kinematics*

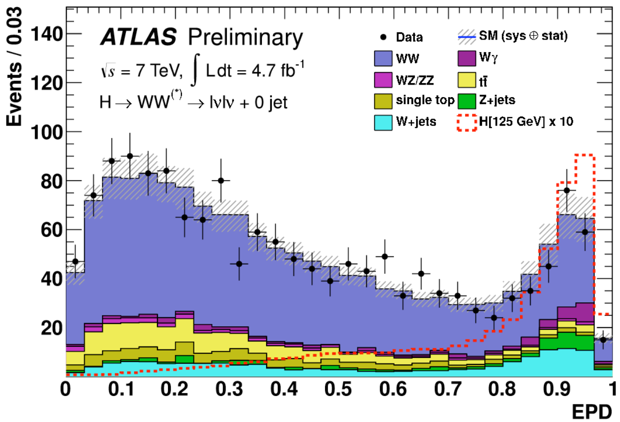
For each possible underlying Process (Matrix Element)

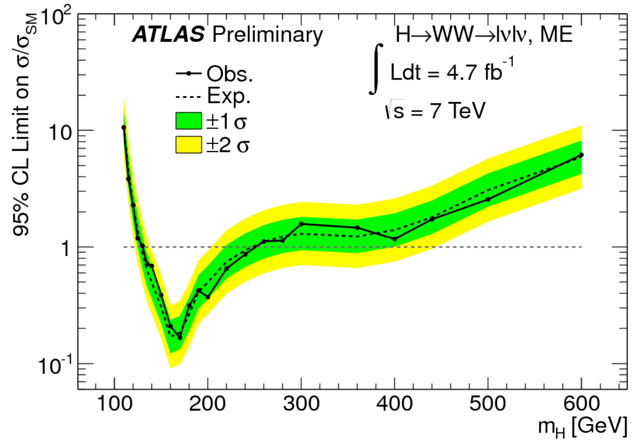
Matrix Element Method at the LHC

ATLAS has applied the Matrix Element Method to H→WW searches

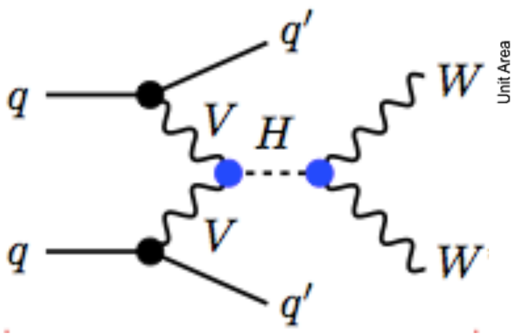


gluon-gluon fusion

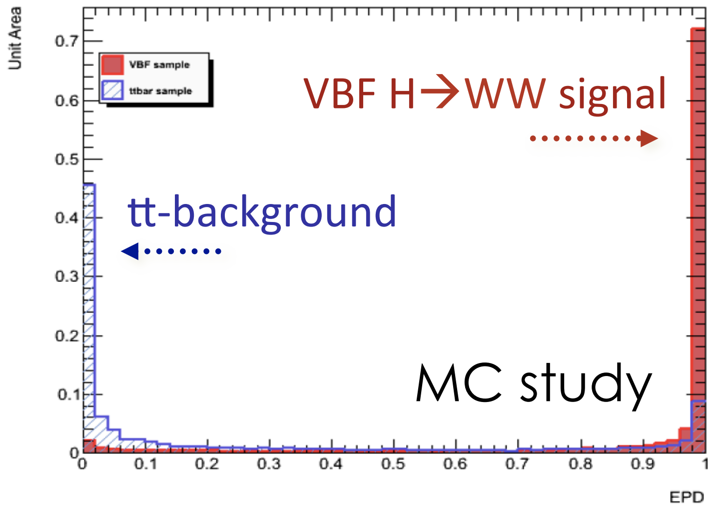




most sensitivity gain observed for high Higgs mass



vector boson fusion



Discriminant based on ratio of probability densities

$$EPD = \frac{P_{Signal}}{P_{Signal} + P_{Background}}$$

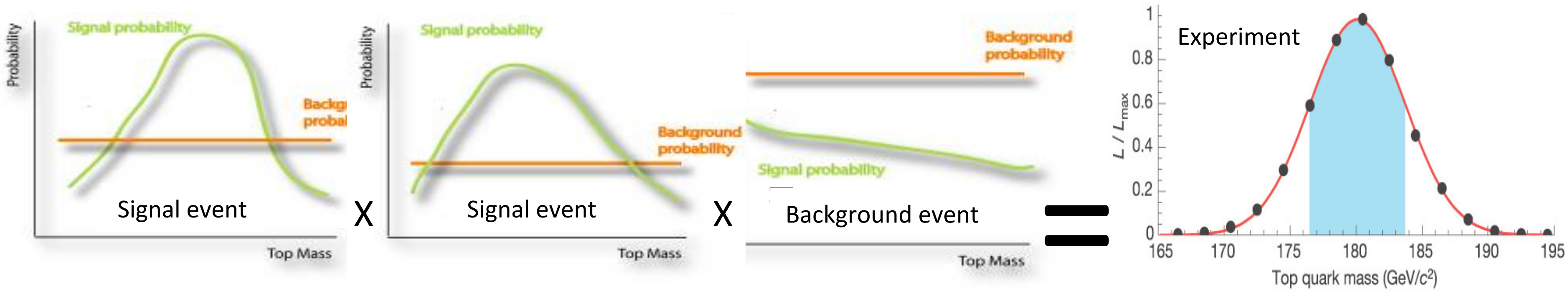
Matrix Element Method for precision top quark mass measurement:

– First application of the method at the Tevatron in 2004

– Evaluate ttbar probability densities as a function of M_{top}

$$L(c_s, M_{top}) \propto \prod_{i=1}^{N} (c_s\, P_{ttbar,i}(x, M_{top}) + (1 - c_s)\, P_{W+jets,i}(x))$$

– Obtain best value for M_{top} by multiplying event probability densities:

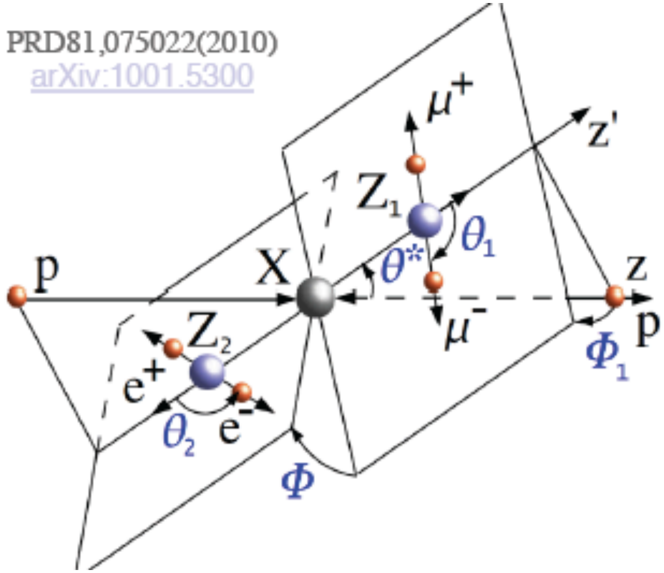


Nature 429, 638-642 (2004)

Matrix Element for Property Measurements

Matrix Element Method for Higgs searches + property measurements

PRD81,075022(2010)
arXiv:1001.5300

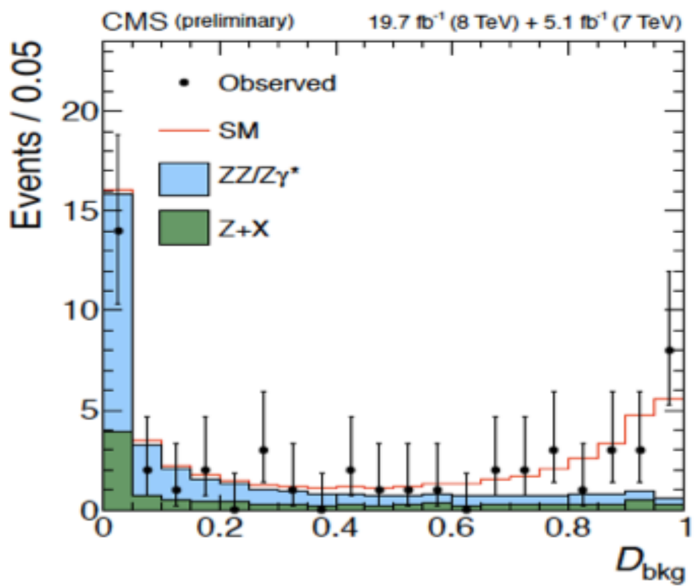


Matrix Element Likelihood Analysis:
uses kinematic inputs for
signal to background discrimination
$\{m_1, m_2, \theta_1, \theta_2, \theta^*, \Phi, \Phi_1\}$

$$\mathrm{MELA} = \left[1 + \frac{\mathcal{P}_{\mathrm{bkg}}(m_1, m_2, \theta_1, \theta_2, \Phi, \theta^*, \Phi_1 | m_{4\ell})}{\mathcal{P}_{\mathrm{sig}}(m_1, m_2, \theta_1, \theta_2, \Phi, \theta^*, \Phi_1 | m_{4\ell})}\right]^{-1}$$

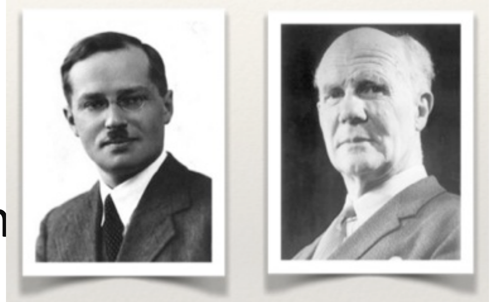
- CMS H→ZZ*→4lepton analysis
- Clean analysis, only well measured leptons no unobserved particles,
  → No phase space integration
- Easy access to property measurements, e.g.

$$\boxed{\mathrm{pseudoMELA} = \frac{P_{0+}}{P_{0+} + P_{0-}}}$$



Pros and Con of Matrix Element Technique

PROS:



$$\Lambda(x) = \frac{L(\theta_0 \mid x)}{L(\theta_1 \mid x)}$$

- The method provides access to most powerful test statistic, the *likelihood ratio*, for discriminating between alternative hypotheses - Neyman-Pearson lemma
- The method 'knows already all the physics'
- The probability densities P_i that fully characterize candidate events directly *depends* on the physical parameters of interest (allows to scan the physics parameters – e.g. couplings or masses)
- It avoids tuning on unphysical parameters for analysis optimization
- It requires no training, thereby mitigating dependence on large samples of simulated events and MC modeling issues

CONS:

- Performing phase space integration can be computationally prohibitive

CPU versus GPU

- Traditionally, the Matrix Element Analysis has been performed on CPUs
- The computational issue:
  - Complex final states with unobserved particles (e.g. neutrinos) or many poorly measured objects (e.g. jets)
  - Requires *multi-dimensional phase space integral* for each event
  - Easily use up many thousands of CPU hours for a typical analysis (slows down optimization and debugging)
- A solution:
  - Graphics Processing Units (GPUs) provide cheap, extensive parallel processing
  - Need to implement Matrix Element analysis on GPUs
  - Ideally, the implementation should be generic and flexible to accommodate any particle physics process

Monte Carlo Phase Space Integration

- Many processes will require ≥ 3 dimensional phase space integrals (moreover when considering higher order effects, e.g. NLO recoil)
- For dim ≥ 3, Monte Carlo Integration is the natural solution
- Employ VEGAS algorithm (stratified and importance sampling technique)

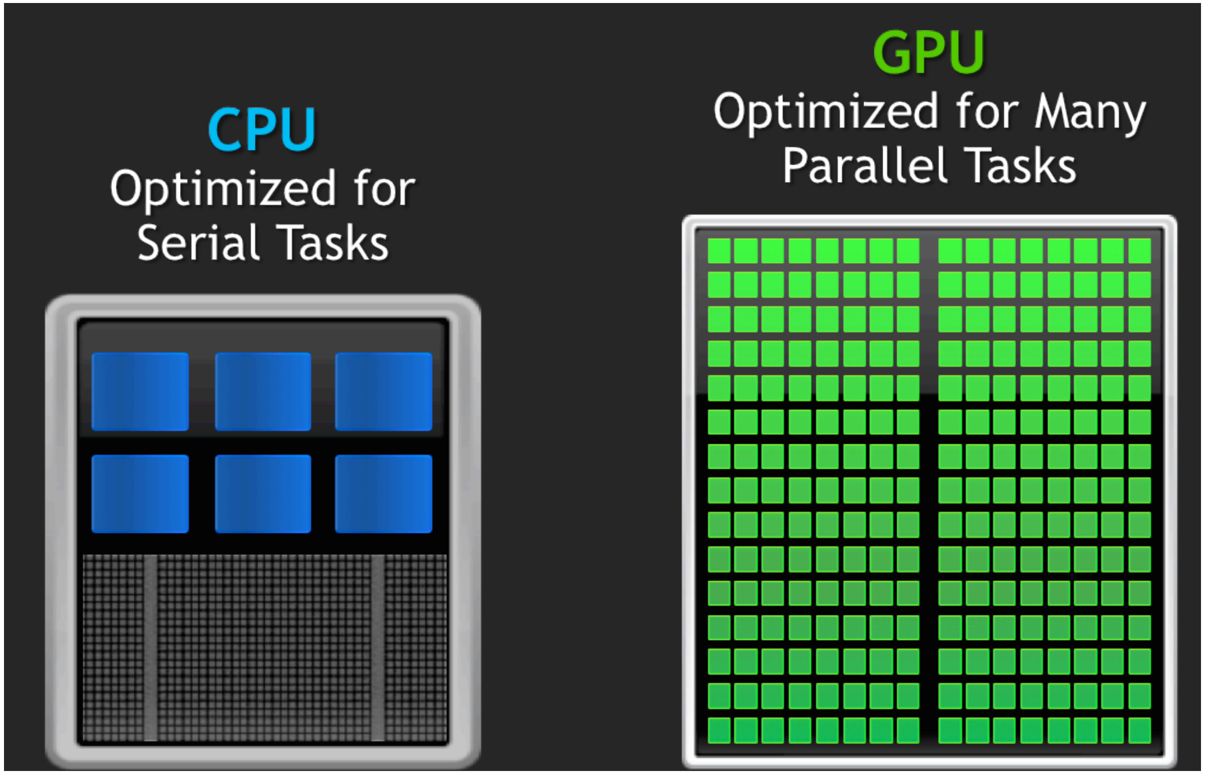
$$I = \int_{V_m} f(\vec{x})\, d\vec{x} \longrightarrow S_N \equiv V_m \underbrace{\frac{1}{N}\sum_{i=1}^{N} f(\vec{x}_i)}_{\equiv \bar{f}}$$

Excellent candidate for parallel computing

Residual error after evaluating N points

$$\Delta S_N \approx \frac{V_m}{\sqrt{N}} \underbrace{\left(\frac{1}{N-1}\sum_{i=1}^{N}(f(\vec{x}_i) - \bar{f})^2\right)}_{\equiv \sigma_f}$$

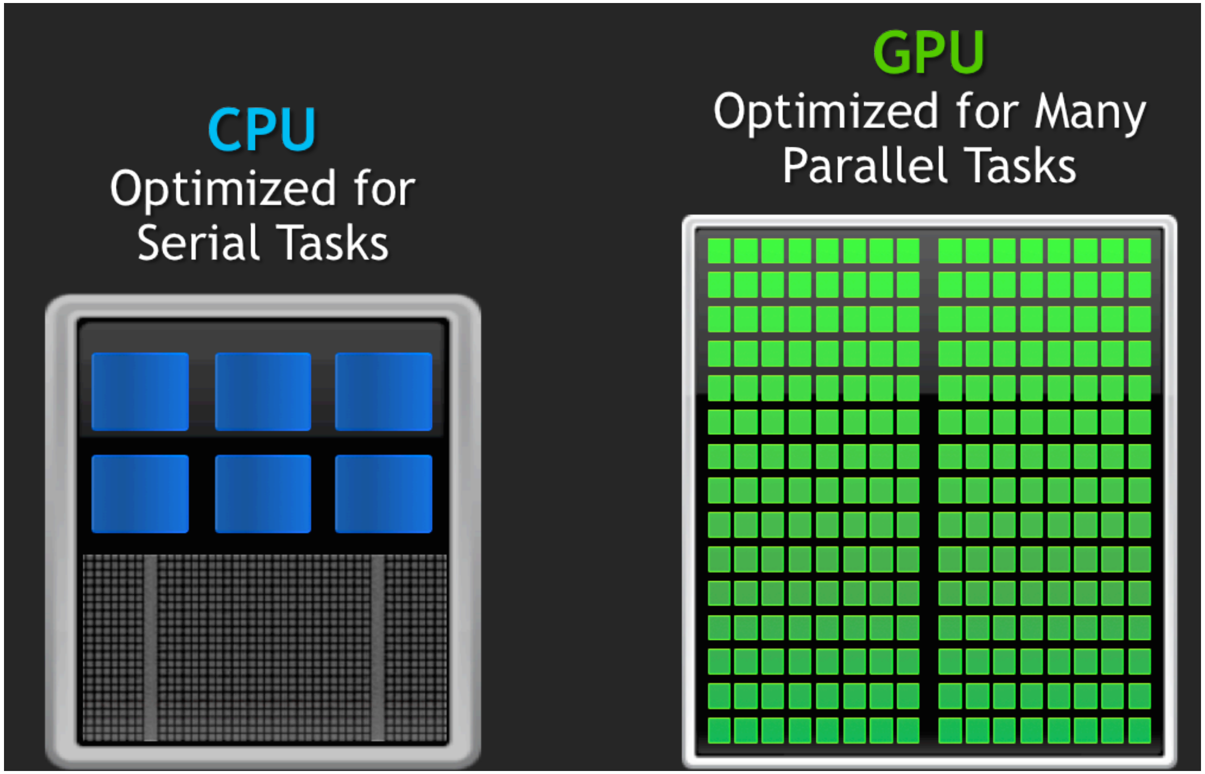
CPU vs GPU for Matrix Element Calculation



C++, Fortan, Python

CUDA, OpenCL

CPU vs GPU for Matrix Element Calculation



Data parallelism in the Matrix Element Method maps well to the *single instruction multiple data architecture* of GPUs

- The function evaluation is the *single Instruction* for every core
- Each phase space point to be evaluated is the *multiple data*

C++, Fortan, Python

OpenCL, CUDA

→ Develop plugin for MadGraph to write out code for both architectures

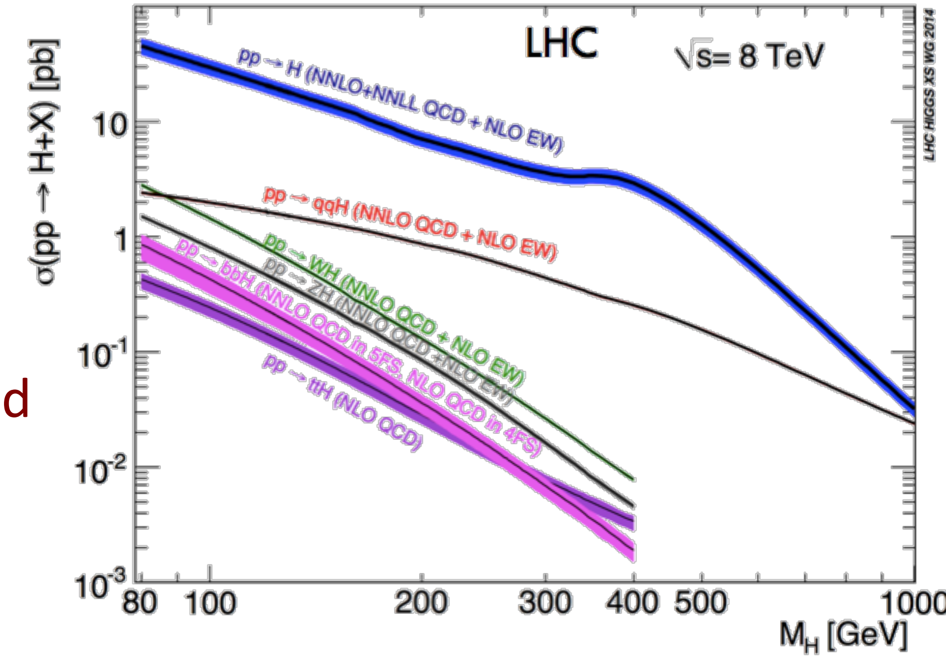
A GPU ME Plugin for MadGraph

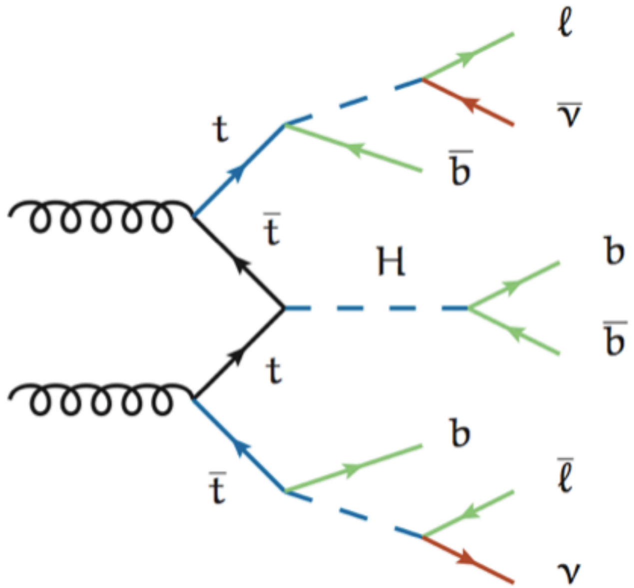
```
************************************************************
*                                                          *
*          W E L C O M E  to  M A D G R A P H  5          *
*                                                          *
*                                                          *
*                 *                       *                *
*                   *        * *        *                  *
*                     * * * * 5 * * * *                    *
*                   *        * *        *                  *
*                 *                       *                *
*                                                          *
*        VERSION 1.5.14               2013-11-27           *
*                                                          *
*    The MadGraph Development Team - Please visit us at    *
*    https://server06.fynu.ucl.ac.be/projects/madgraph     *
*                                                          *
************************************************************
mg5>
mg5>generate p p > h j j  $$ w+ w- / z ,  h > w+ w- >  e+ ve mu- vm~
...
mg5>output standalone_ocl me_vbf_ocl
...
Output to directory /home/stelzer/MadGraph5_v1_5_14_mod/bin/me_vbf_ocl done.
mg5>exit
```

- Developed an easy to use plug-in for MadGraph
- Interface similar to MadWeight
- Export skeleton code for *any* 2→N particle process in CUDA, OpenCL & C++

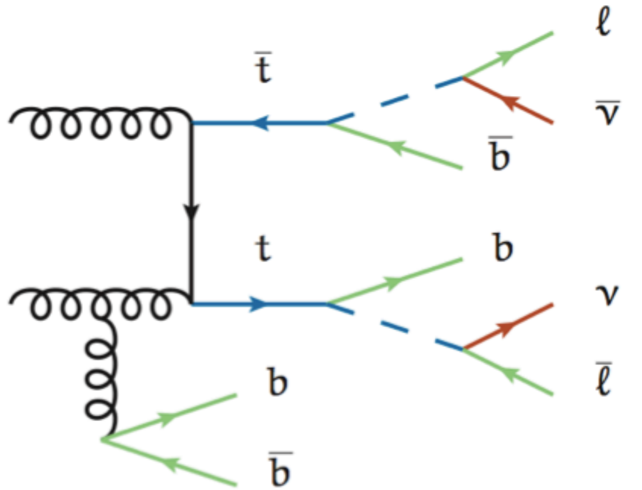
A Search for ttH - GPU Case Study

- Observation of tt + Higgs *will* provide direct access to top Yukawa coupling
- Combinatorial background for M_{bb} reconstruction is large given 4-6 jets
- Complex final state provides a good benchmark for Matrix Element method
- *One of the highlights of LHC Run 2*
  → *Cross section will increase by x4.7*



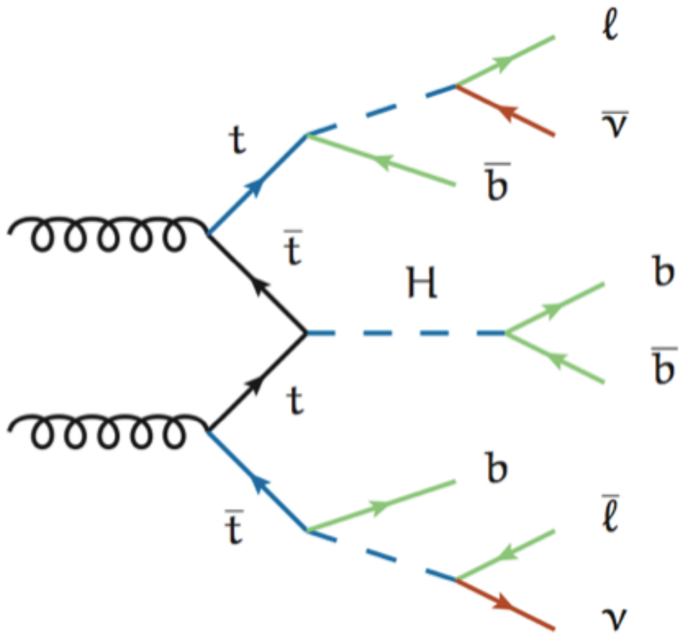


ttH - signal



tt+bb Irreducible background

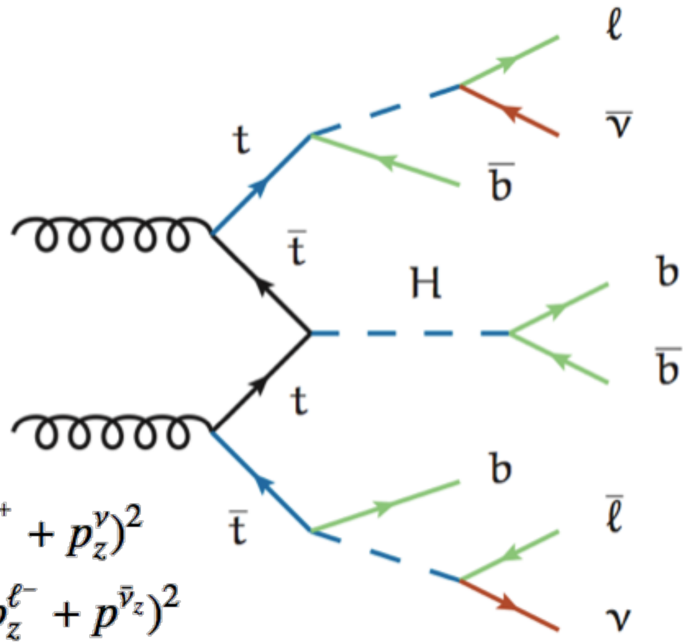
Dimension of Phase Space Integral



- Independent kinematic variables:
  - The 8 *final state particles* correspond to 8 x 3 = 24 degrees of freedom
  - The longitudinal momenta of the *initial state partons amount to* 2 degrees of freedom
  - Total: 26 degrees of freedom
- Constraints:
  - Lepton momenta are well measured (i.e their TFs delta functions): 3 x 2 = 6 constraints
  - Jet direction are well measured: 2 x 4 = 8 constraints
  - Energy and momentum conservation: 4 constraints
  - Total: 18 constraints
- Phase space integral:
  - 26 degrees of freedom – 18 constraints = 8 dimensional integral
  - E.g. Integration over: $E_{b1}, E_{b2}, E_{b3}, E_{b4}, p^x_{\nu 1}, p^y_{\nu 1}, p^z_{\nu 1}, p^z_{\nu 2}$

Dimension of Phase Space Integral

- The phase space has sharp peaks corresponding to the on shell propagators.
- The phase space integral can be simplified for the numerical integrator by an appropriate variable transformation

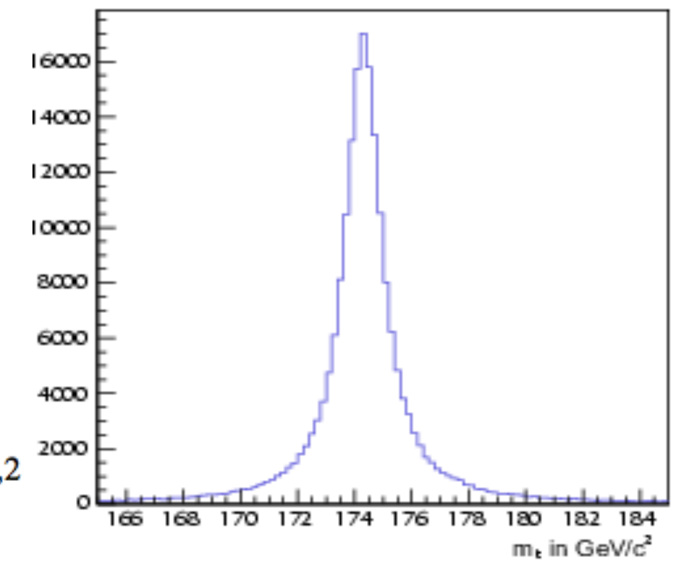


for tt system

$$
\begin{aligned}
q^2_{W^+} &= (E_{\ell^+} + E_\nu)^2 - (p_x^{\ell^+} + p_x^\nu)^2 - (p_y^{\ell^+} + p_y^\nu)^2 - (p_z^{\ell^+} + p_z^\nu)^2 \\
q^2_{W^-} &= (E_{\ell^-} + E_{\bar\nu})^2 - (p_x^{\ell^-} + p^{\bar\nu_x})^2 - (p_y^{\ell^-} + p^{\bar\nu_y})^2 - (p_z^{\ell^-} + p^{\bar\nu_z})^2 \\
q^2_t &= (E_b + E_{\ell^+} + E_\nu)^2 - (p_x^b + p_x^{\ell^+} + p_x^\nu)^2 - \\
&\quad (p_y^b + p_y^{\ell^+} + p_y^\nu)^2 - (p_z^b + p_z^{\ell^+} + p_z^\nu)^2 \\
q^2_{\bar t} &= (E_{\bar b} + E_{\ell^-} + E_{\bar\nu})^2 - (p_x^{\bar b} + p_x^{\ell^-} + p_x^{\bar\nu})^2 - \\
&\quad (p_y^{\bar b} + p_y^{\ell^-} + p_y^{\bar\nu})^2 - (p_z^{\bar b} + p_z^{\ell^-} + p_z^{\bar\nu})^2,
\end{aligned}
$$

H only

$$
\begin{aligned}
f &= (E_1 + E_2) \\
m_H^2 &= (E_1 + E_2)^2 - |\vec{p}_1|^2 - |\vec{p}_2|^2 - 2\,|\vec{p}_1|\,|\vec{p}_2| \cos\Delta\theta_{1,2}
\end{aligned}
$$



Matrix Element Method for ttH Search

Event probability density signal and background hypothesis:

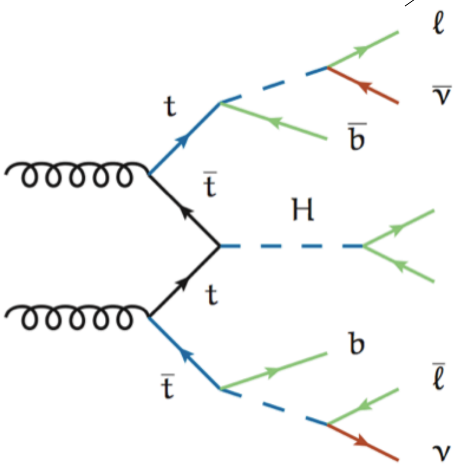
Matrix Element

Parton distribution functions

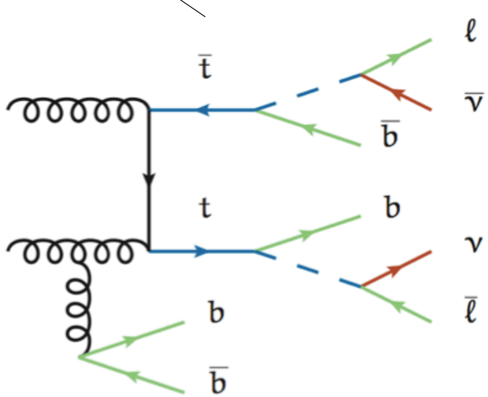
Integration over part of the phase space Φ

$$\hat{p}_i = \frac{1}{\sigma_i}\sum_{jet\ comb}\sum_{flavour}\overbrace{\int |M(p_i)|^2}\frac{f(x_1)f(x_2)}{|q_1|\cdot|q_2|}\phi\, dp^3_{\nu 1}\, dp^z_{\nu 2}\prod_{j=1}^{N_{jet}=4} TF(E_j^{jet};E_j^{part})\cdot dE_j$$

Transfer Function



Signal



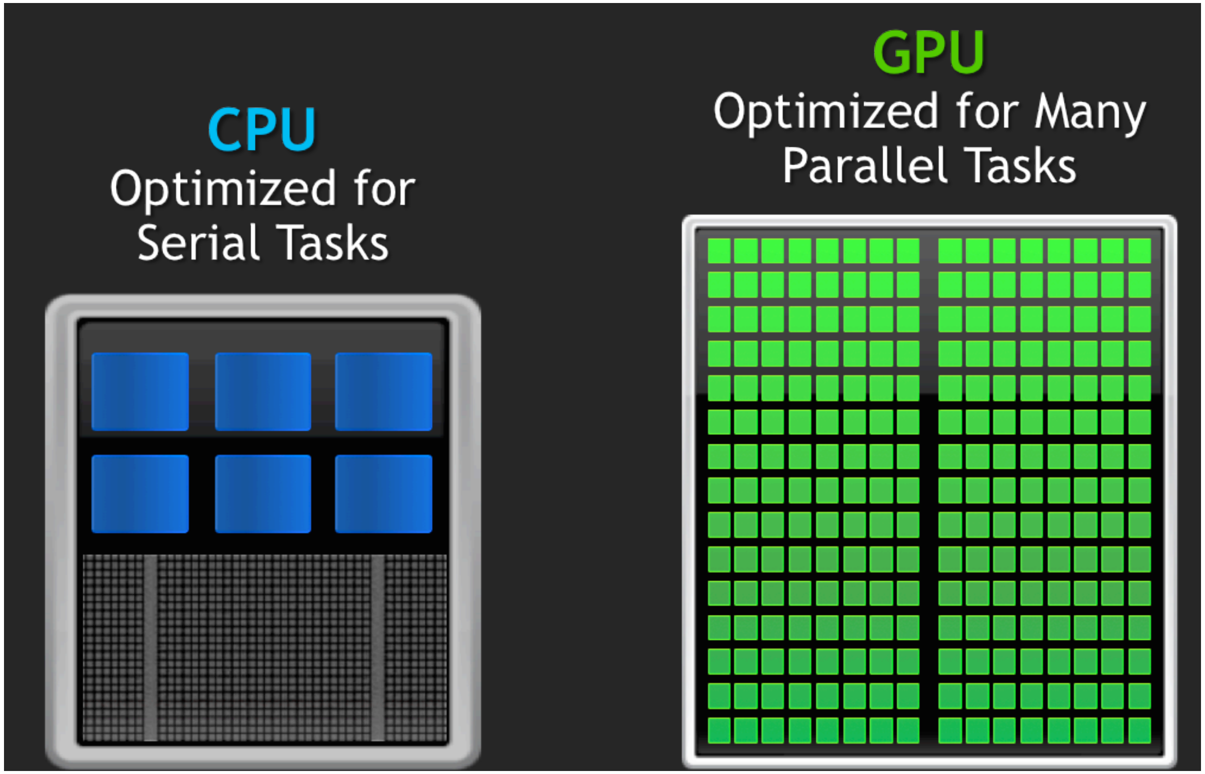
Background

Construct discriminant

$$D = \log_{10}\left(\frac{\hat{p}_{t\bar{t}H}}{\hat{p}_{t\bar{t}b\bar{b}}}\right)$$

- Evaluation of the integrand is broken into components for the matrix element *M*, the PDF's, the TF's and the phase space ϕ – a single GPU "kernel" program each
- PDF data is stored in (x, Q^2) grids for each parton flavor and passed to the kernel.

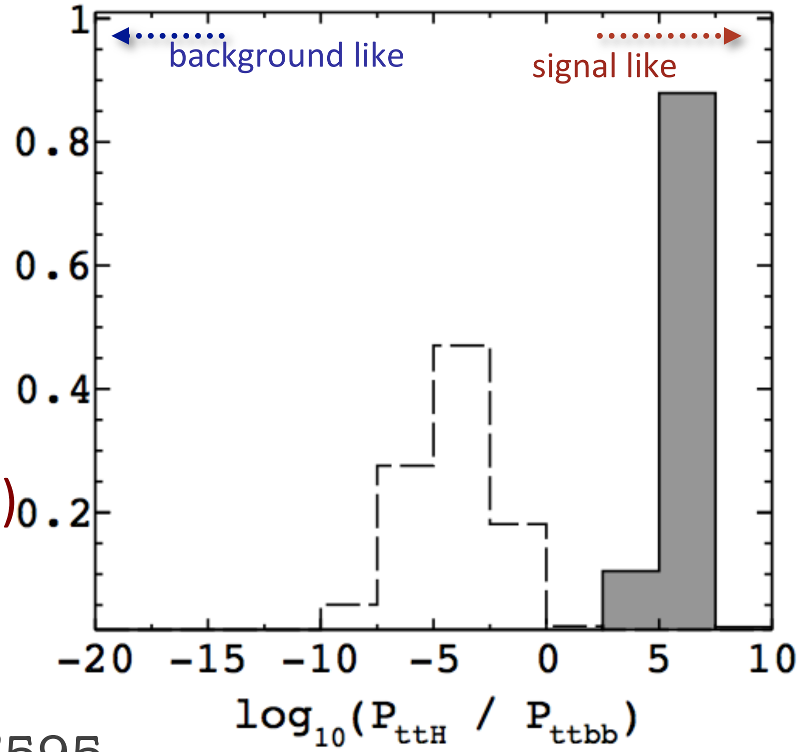
Current Hardware



Configuration	Details
CPU	Intel Xeon CPU E5-2620 0 @ 2.00GHz (single core) using gcc 4.8.1
CPU (MP)	Intel Xeon CPU E5-2620 0 @ 2.00GHz (twelve cores + hyperthreading) using AMD SDK 2.9 / OpenCL 1.2
GPU	AMD Radeon R9 290X GPU (2,816 c.u.) using AMD SDK 2.9 / OpenCL 1.2
GPU_x	Same hardware as GPU but with modifications to the code to accomodate GPU architecture

Particle Level Results

- First we compute the event probability densities assuming only δ-functions for the Transfer Functions
- I.e. only considering jet-parton combinatorics but no phase space integration
- Large improvement for GPU (~100-400x)



$$D = \log_{10}\left(\frac{\hat{p}_{t\bar{t}H}}{\hat{p}_{t\bar{t}b\bar{b}}}\right)$$

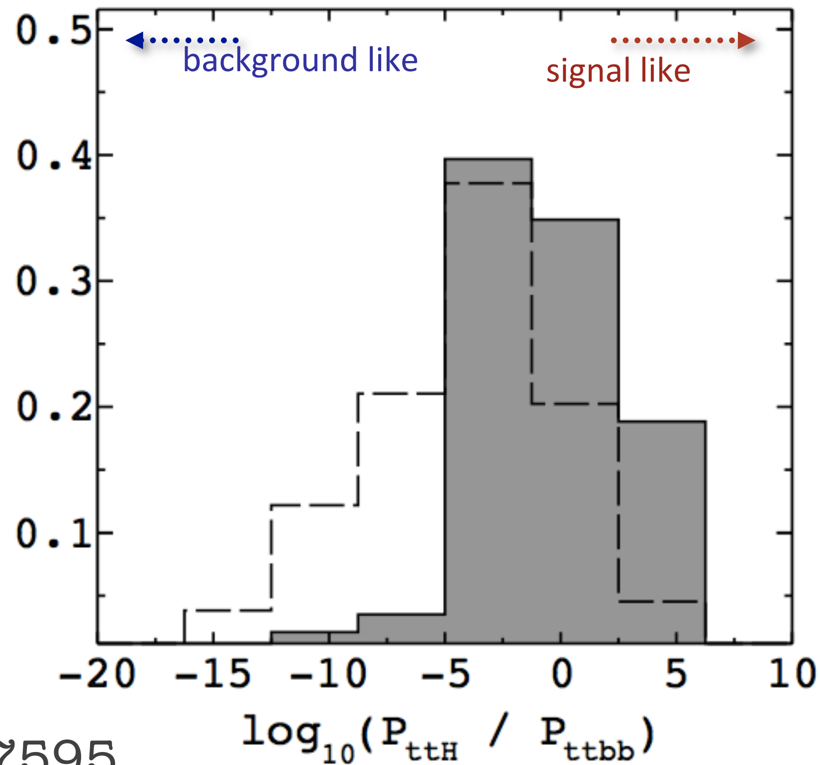
arXiv:1407.7595

Parton Speed-up: Time in Seconds for 100,000 Events

Process	CPU	CPU (MP)	GPU	GPU_x	GPU_x/CPU
signal	255	29	1.8	0.7	364
background	661	91	12	5.4	122

Hadron Level Results

- Assuming δ-functions for lepton and jet directions, use Transfer Functions jets (derived from hadron level jets)
- I.e. evaluation of 8-dimensional phase space integration for each event
- Improvement for GPU / CPU (~50x)



$$D = \log_{10}\left(\frac{\hat{p}_{t\bar{t}H}}{\hat{p}_{t\bar{t}b\bar{b}}}\right)$$

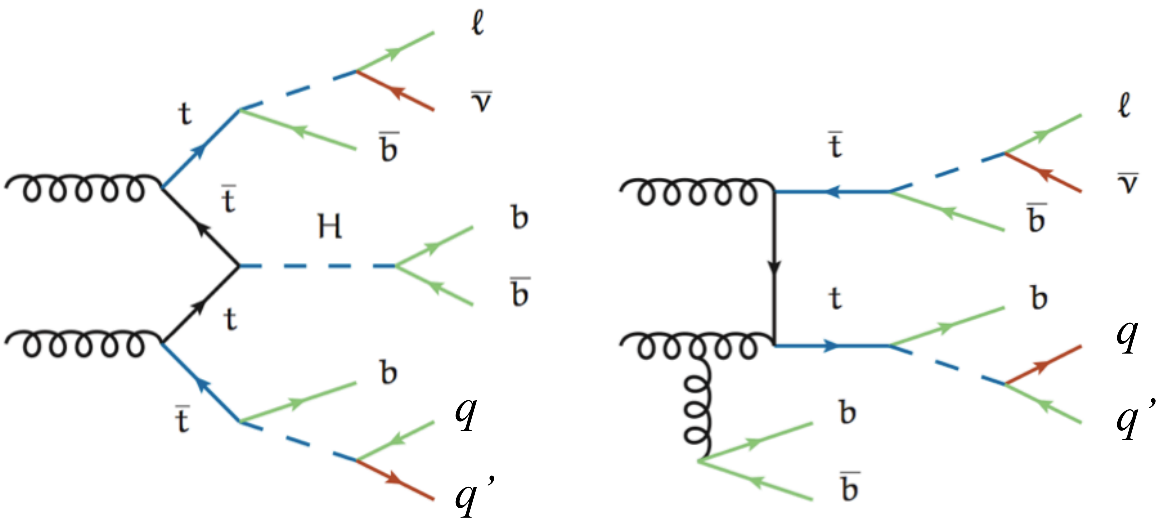
arXiv:1407.7595

Hadron Speed-up: Time in Seconds for 1 Event

Process	CPU	CPU (MP)	GPU	GPU_x	GPU_x/CPU
signal	312	36.2	7.5	5.9	52.0
background	405	55.1	9.1	7.1	57.3

Hadron Level Results

- Recently implemented ttH analysis in lepton + jets channel



- Only 6-dim phase space integral (instead of 8-dim)
- Variable transformation much simpler
- More jet-parton permutations
- GPU/CPU speadup >100x

Semi Leptonic Speed-up: Time in Seconds for 1 Event

Process	CPU	CPU (MP)	GPU_x	GPU_x/CPU
signal	1000	91	7.9	125
background	3800 (est.)	347	35	109

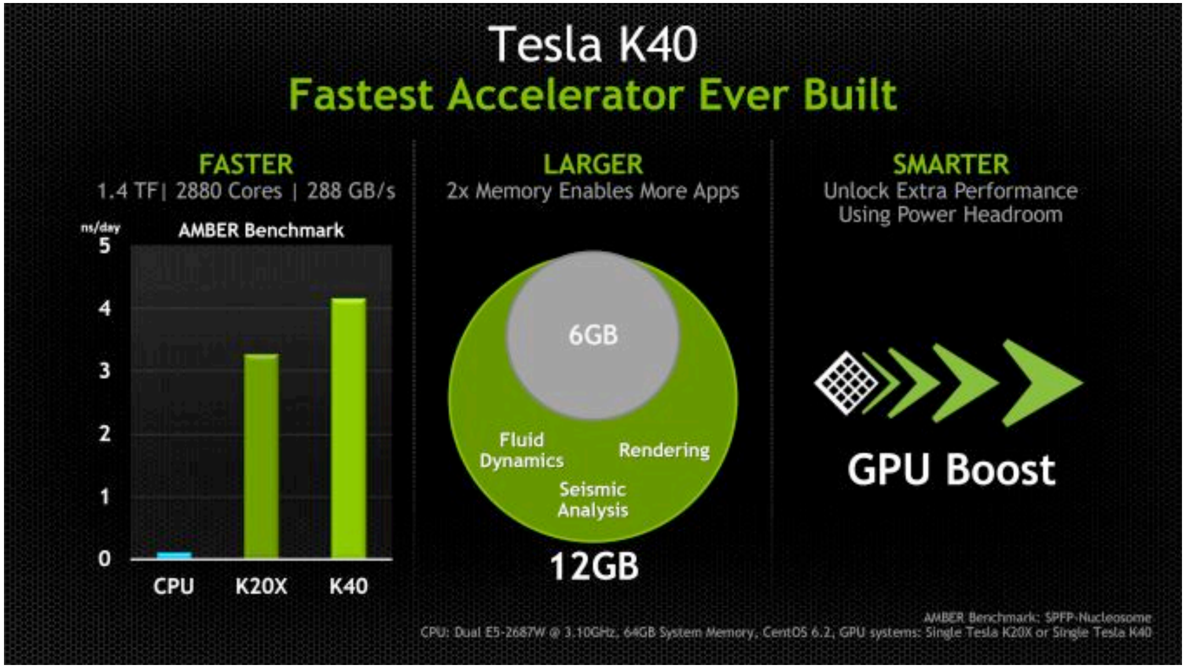
Notes on Hadron Level Performance

- The integration algorithm (VEGAS) is run on the CPU in all cases,
  → Damps the GPU improvements in the integral evaluation.
- Variable transformations and transfer functions add a lot of complexity
  → Requires double floating point precision (slower)
  → Expect larger GPU improvements for simpler processes (as observed)
- The overall duty factor of the GPU is significantly reduced, limited by number of registers to each thread, to as low as 10% - 20%, since the full number of cores is not utilized → Further optimization possible

Next Benchmark Tests

Next steps:

Thanks to the NVIDIA Corporation, we will soon benchmark with NVIDIA Tesla K40

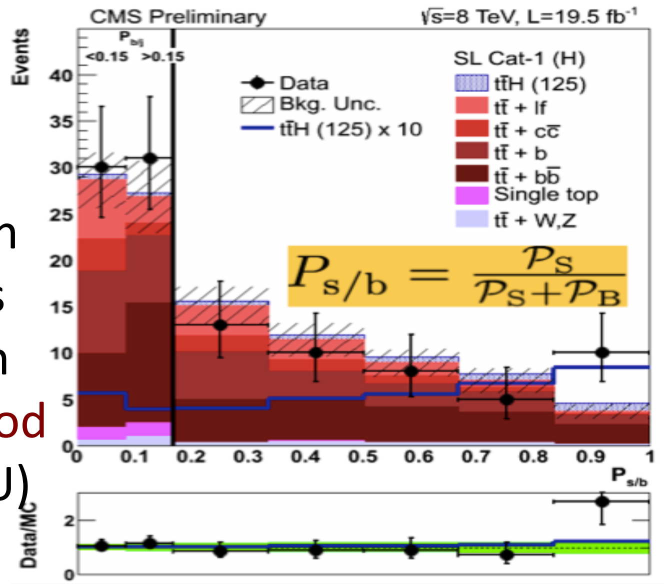


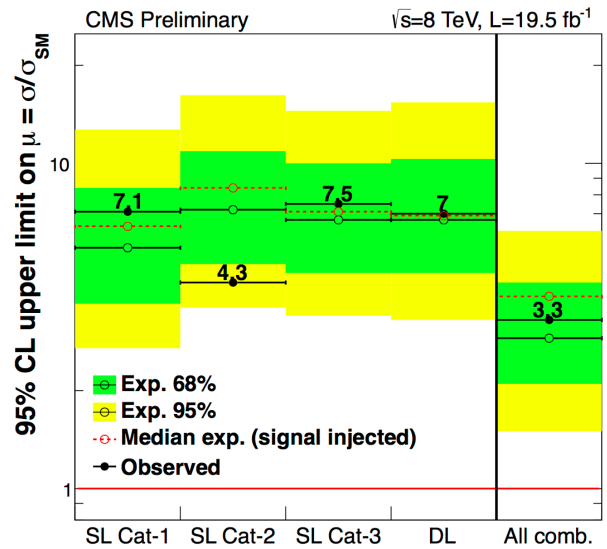
	Nvidia Tesla k40	AMD Radeon R9 290X
Floating-point precision	4.29 Tflops	5.6 Tflops
Bandwidth	288 GB/s	320 GB/s
Memory Size	12 GB	4 GB
Cores	2880	2816
Language	CUDA	OpenCL

Current Results on LHC ttH Searches

CMS ttH Search

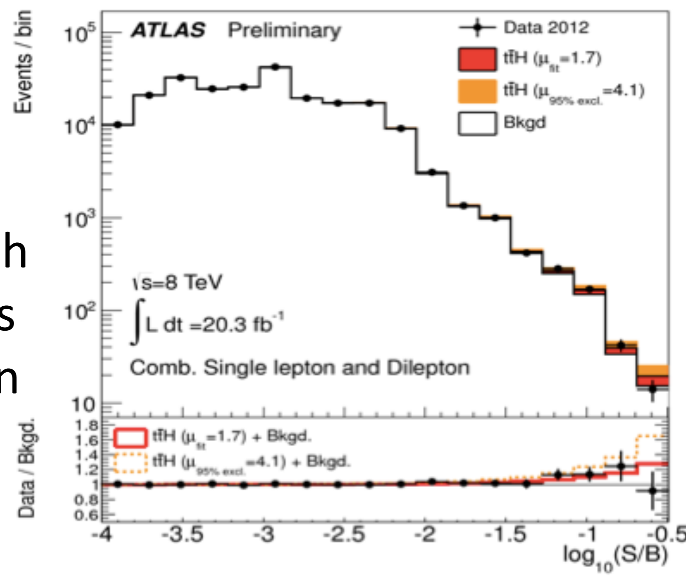
- Dilepton and lepton + jets channel
- 4, 5, ≥6jet sample with 2b, 3b, ≥4b categories
- Discriminant based on Matrix Element Method (still evaluated on CPU)

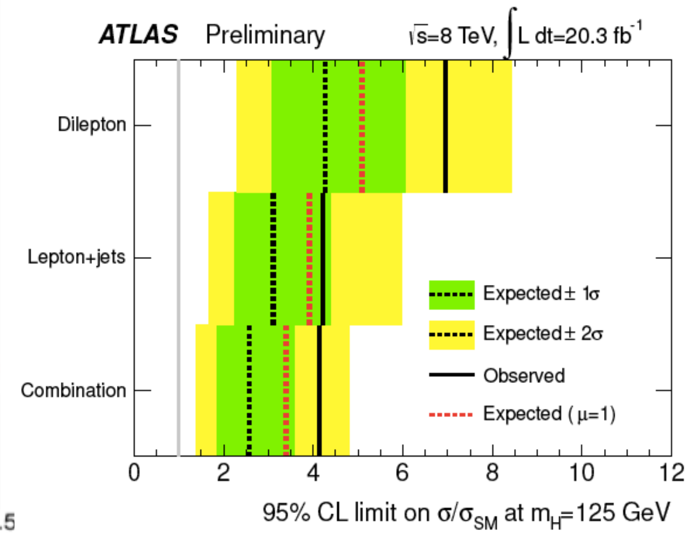




ATLAS ttH Search

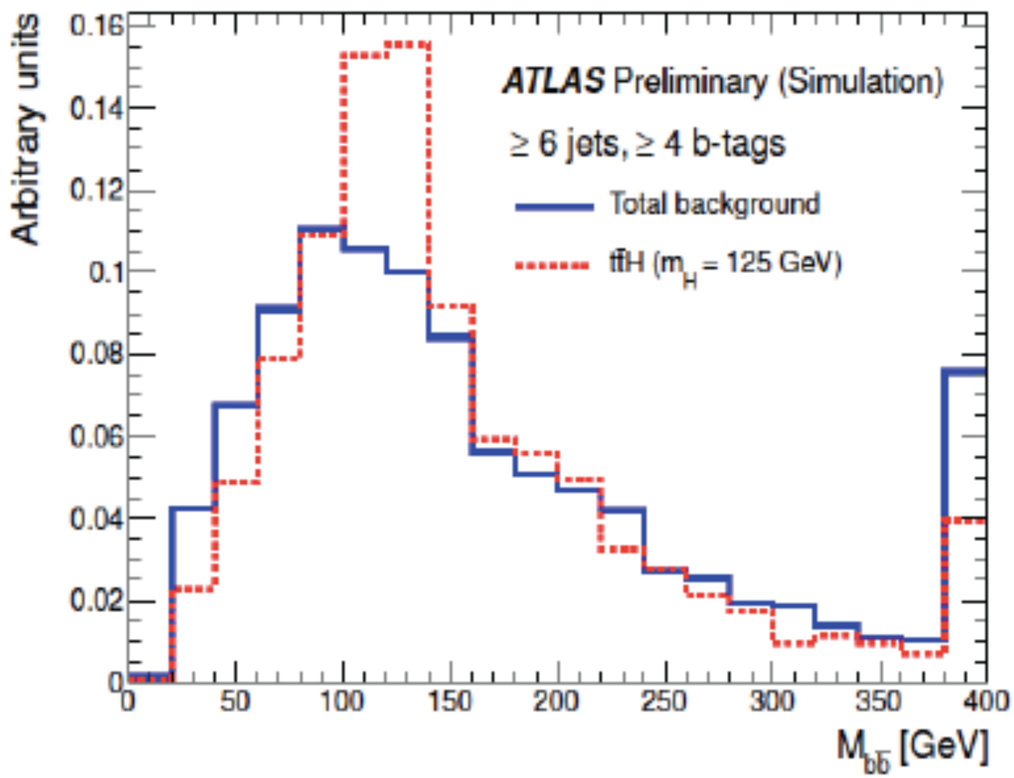
- Dilepton and lepton + jets channel
- 4, 5, ≥6jet sample with 2b, 3b, ≥4b categories
- Discriminant based on Neural Network





Summary

- The computationally prohibitive Matrix Element Method can be sped up by large factors (up to 100+) using GPUs instead of CPUs
- This means, GPU based parallel computing makes the *Matrix Element Method viable for general usage at the LHC*
- We have developed a plugin for the event generator MadGraph, to output GPU compatible MEM code for any 2 → N process
- Useful tool for Higgs measurements, characterization and general searches for new Physics at the LHC
- Short paper submitted to Comp. Phys. Com, arXiv:1407.7595


ATLAS Preliminary (Simulation)
≥ 6 jets, ≥ 4 b-tags
Total background
ttH (m_H = 125 GeV)
Arbitrary units
0.16
0.14
0.12
0.1
0.08
0.06
0.04
0.02
0
0
50
100
150
200
250
300
350
400
$M_{b\bar{b}}$ [GeV]