University of Siegen

Department of Physics

Masterlab

Top Quark Physics at the Large Hadron Collider

Tutor:
PD Dr. Carmen Diez Pardos

Manual:
Sara Ghasemi
(updated by J. Meshreki, C. Diez Pardos)

Top Quark Physics at the Large Hadron Collider

Preface

This experiment introduces the students into data analysis in particle physics. They will gain insight in the Large Hadron Collider (LHC), the particle accelerator at CERN where protons collide with protons at a centre-of-mass energy at the Teraelectron Volts (TeV) scale, and get familiar with the ATLAS and CMS detectors, two of the particle detectors constructed at LHC. With the help of a C++ based framework and ROOT, the students will do simple data analysis, using real and simulated data sets, to investigate the physics of top quark.

Key words: particle physics, high energy physics, LHC, ATLAS, CMS, top quark, data analysis, C++, ROOT

Contents

1 Basics of particle physics

1.1 Fundamental constituents of matter

Particle physics focuses on the study of the basic constituents of matter and their interactions. Our current understanding of particles and their interactions in summarised in the Standard Model (SM) of particle physics.

The elementary particles are categorised according to their spin into *fermions* (half integer spin) or *bosons* (integer spin). The known matter consist of fermions, which are further classified into *quarks* and *leptons*. Their names and some of their properties are summarised in Figure 1.

Quarks and leptons are grouped into three families (also referred to as *generations*), denoted in Figure 1 by Greek numbers. Our visible universe is composed entirely of first-generation particles, namely the up and down quarks, and the electron and its neutrino. Particles in higher generations are heavier and unstable and decay into the first-generation particles, which are stable. However, they can be produced in the high-energy collisions at the particle accelerators (along with the first-generation particles).

The interactions among fermions are mediated by the *bosons*. The electromagnetic force is mediated by the massless photon, the strong force is carried by the massless gluons, and the massive W and Z bosons are responsible for the weak force. Owing to the very small masses of the elementary particles, gravity is negligible compared to the other three forces.

According to the SM, particles acquire their mass through interaction with the Higgs field. The existence of this field was proven by the discovery of the Higgs boson, announced in 2012 by the ATLAS and CMS experiments.

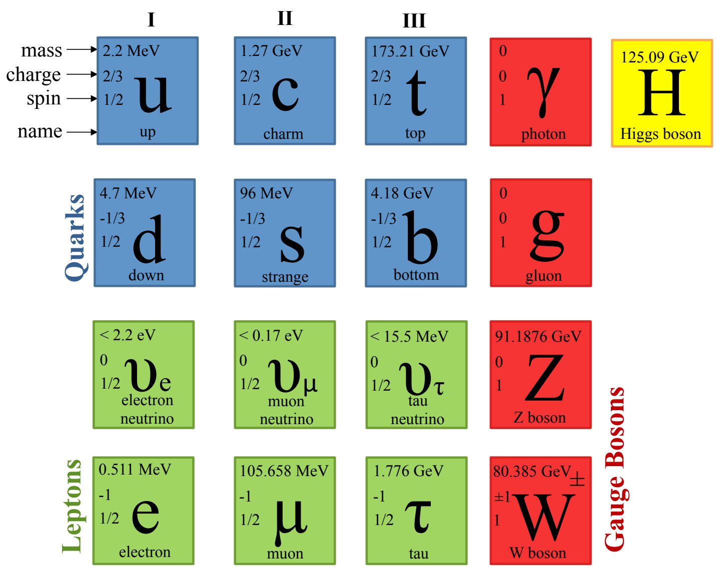

Figure 1: Elementary particles according to the SM of particles physics.

1.2 Relativistic kinematics

According to Einstein's theory of relativity, energy and mass can be converted to each other:

$$E = mc^2 \,.$$

The relativistic mass is a function of the velocity:

$$m = \frac{m_0}{\sqrt{1-\frac{v^2}{c^2}}} \,,$$

so the energy of a particle and its momentum can be written as:

$$E = \frac{m_0 c^2}{\sqrt{1-\frac{v^2}{c^2}}} \quad , \quad \vec{p} = \frac{m_0 \vec{v}}{\sqrt{1-\frac{v^2}{c^2}}} \,.$$

The energy-momentum four-vector of a particle, $\boldsymbol{p}$, can be expressed as:

$$\boldsymbol{p} = \begin{pmatrix} E/c \\ p_x \\ p_y \\ p_z \end{pmatrix} = \begin{pmatrix} E/c \\ \vec{p} \end{pmatrix} \,.$$

In particle collisions and the decay of particles $\boldsymbol{p}$ is conserved.

The scalar product of $\boldsymbol{p}$ with itself gives the square of the rest mass of the particle:

$$\boldsymbol{p}^2 = (E/c)^2 - ({p_x}^2 + {p_y}^2 + {p_z}^2) = {m_0}^2 \,.$$

The invariant mass of two colliding particles, or a two-particle decay, is:

$$M_{12}^2 = (\boldsymbol{p_1} + \boldsymbol{p_2})^2 = \begin{pmatrix} E_1/c + E_2/c \\ p_{x_1} + p_{x_2} \\ p_{y_1} + p_{y_2} \\ p_{z_1} + p_{z_2} \end{pmatrix}^2 \,.$$

In particle physics, it is common to use the natural unit system[1], where the constant c can be omitted. Hereafter, we will also use the natural units.

Example 1: The centre-of-mass energy of two protons with same energies in a head-on collision can be calculated as:

$$E_{c.m.}^2 = s = (\boldsymbol{p_1} + \boldsymbol{p_2})^2$$
$$= 2{m_0}^2 + 2E^2 - 2\vec{p}.(-\vec{p})$$

In a high energy collider, the proton mass would be negligible compared to its energy ($m_0 \ll E$), so one could make use of $m_0 \approx 0$ and $|\vec{p}| \approx E/c$ and write:

[1]In natural units $c = \hbar = k_B = 1$, therefore [energy] = [mass] = [1/length] = [1/time] = Electron Volt (eV).

$$E^2_{c.m.} = s = 4E^2$$
$$E_{c.m.} = \sqrt{s} = 2E$$

Example 2: The square of invariant mass of two colliding particles, or a two-particle decay, in a high energy collider can be calculated as:

$$M^2_{12} = (\boldsymbol{p_1} + \boldsymbol{p_2})^2 = \boldsymbol{p_1}^2 + \boldsymbol{p_2}^2 + 2\,\boldsymbol{p_1}.\boldsymbol{p_2}$$
$$= {m_1}^2 + {m_2}^2 + 2\,(E_1E_2 - |\vec{p_1}||\vec{p_2}|\cos\theta_{12})$$

where θ_{12} is the angle between $\vec{p_1}$ and $\vec{p_2}$. Using $m_{1,2} \ll E_{1,2}$, one gets:

$$M^2_{12} = 2\,E_1E_2(1 - \cos\theta_{12})$$

In particle collider experiments, the angular positions of the particles are usually measured in terms of the azimuthal angle, ϕ, and the *pseudorapidity*, which is defined as

$$\eta = -\ln(\tan\frac{\theta}{2})\,,$$

where θ is the polar angle. Also, usually only the transverse momentum of the particle $p_T = \sqrt{{p_x}^2 + {p_y}^2}$ is well measured, but not its momentum. So it is convenient to write M^2_{12} in terms of ϕ, η and p_T:

$$M^2_{12} = 2\,p_{T_1}p_{T_2}(\cosh(\eta_1 - \eta_2) - \cos(\phi_1 - \phi_2))$$

1.3 Energy units

In particle physics energy is expressed in Electron Volts (eV) instead of Joules. One Electron Volt is the kinetic energy of a single electron when moving through an electric potential of 1 Volt. This is equal to $1.602\times10^{-19}J$.

In high energy regimes, we deal with energies of the order of several million Electron Volts, so it is convenient to use MeV, GeV and TeV.

1.4 Proton-Proton interactions at the Terascale

Protons are not elementary particles. They are bound states of three quarks: two up quarks and one down quark. These are called the *valence quarks*. Additionally, the protons are formed by gluons that bound the quarks together and a sea of quarks and antiquarks. All the contents of the proton are called *partons*.

At high energies, for example at the LHC machine, the partons of the incoming protons interact with each other and not the protons themselves.

Each parton carries a fraction x of the momentum of the proton, $\boldsymbol{p_P}$. Therefore, the energy available in the collision is determined by the four-momentum of the partons (p_1, p_2):

$$\hat{s} = (\boldsymbol{p_1} + \boldsymbol{p_2})^2$$

$$\boldsymbol{p_1} = x_1\, \boldsymbol{p_P} \quad \boldsymbol{p_2} = x_2\, \boldsymbol{p_P}$$

$$\hat{s} = s\, x_1\, x_2$$

, where $\sqrt{s}$ is the centre-of-mass energy of the proton-proton collision.

1.5 Hadrons

Hadron is a general name for any composite particle that is a bound state of quarks. Protons, neutrons and pions are some examples of hadrons.

1.6 Jets

The scattered quarks, antiquarks or gluons can never be observed as free particles, instead they are confined in hadrons, forming a spray of hadrons. This spray is called "jet". This is a consequence of the fact that quarks and gluons undergo strong interactions (while the other known particles are not), which have a special structure.

A jet can contain from one to few tens of hadrons. The initial energy of the parton that has initiated the jet is shared among the hadrons in the jet. Typically only a small amount of the initial energy is used for forming the new hadrons and most of the initial energy is carried in the hadrons motion energy. Therefore a jet flies approximately in the same direction of the parton that initiated it.

1.7 Luminosity

Luminosity gives a measure of how many collisions happened in the particle accelerator over a period of time. It depends on the setting of the particle collider machine and is given in inverse area units.

1.8 Cross section

In particle physics a cross section corresponds to the probability that two particles interact in a certain way. For instance, the cross section of the top-antitop quark pair production in proton-proton collisions means the probability that the two protons collide and create a pair of top-antitop quarks ($pp \to t\bar{t}$). In practice, when we measure this cross section, we basically count how many top-antitop pairs are created in the number of proton-proton collision that occurred.

Cross section is denoted by σ and measured in units of area. The SI unit of σ is m^2 but in particle physics the conventional unit is *"barn"*(b):

$$1\mathrm{b} = 10^{-28}\ \mathrm{m}^2\,.$$

The following expression relates the luminosity and the cross section:

$$\sigma = \frac{N}{L} ,$$

where N is the number of times that the interaction we are interested in has happened (for example, number of $t\bar{t}$ pairs) and L is the Luminosity.

Usually a detector is not 100% efficient and it does not cover the whole phase space. So, in practice the total cross section is calculated as:

$$\sigma = \frac{N}{A.C.L} , \quad (1)$$

where A is called the *acceptance* factor and C the *efficiency* factor.

The acceptance factor corrects for the fact that we only measure a portion of the whole phase space, which is called the *fiducial region*. The efficiency factor corrects for the fact that not all interactions inside the fiducial region are detected by the detector.

More on acceptance and efficiency is written in the assignments section.

2 The experiment environment

In this lab course you will analyse the data recorded by the CMS detector from proton-proton collisions at a centre-of-mass energy of $\sqrt{s} = 7$ TeV at the LHC.

2.1 The Large Hadron Collider (LHC)

The LHC is the world's largest particle accelerator, built at the European center for particle physics, CERN [2]. It is a circular accelerator that lies in a tunnel of 27 kilometres in circumference under the ground, at an average depth of around 100 meters [3]. It is located beneath the France-Switzerland border, near the city Geneva. In Figure 2, an aerial view of the region under which the LHC is located is shown.

The LHC accelerates two beams of protons circulating in opposite directions. This is done by two different beam pipes and two independent magnet systems. There are four crossing points where these two beams are brought to collide. At the beginning, the LHC was colliding protons at a centre-of-mass energy of $\sqrt{s} = 7$ TeV (in 2010 and 2011), which then was increased to 8 TeV (in 2012), and to 13 TeV (2015-2018).

At each of the four interaction points a particle detector is positioned. The four main detectors can be seen in Figure 3. ATLAS and CMS are the two general-purpose particle detectors, while the others are designed for specific studies.

[2] **C**onseil **E**uropéen pour la **R**echerche **N**ucléaire.

[3] Its depth varies between 175 m (under the Jura Mountains) and 50 m (towards Lake Geneva).

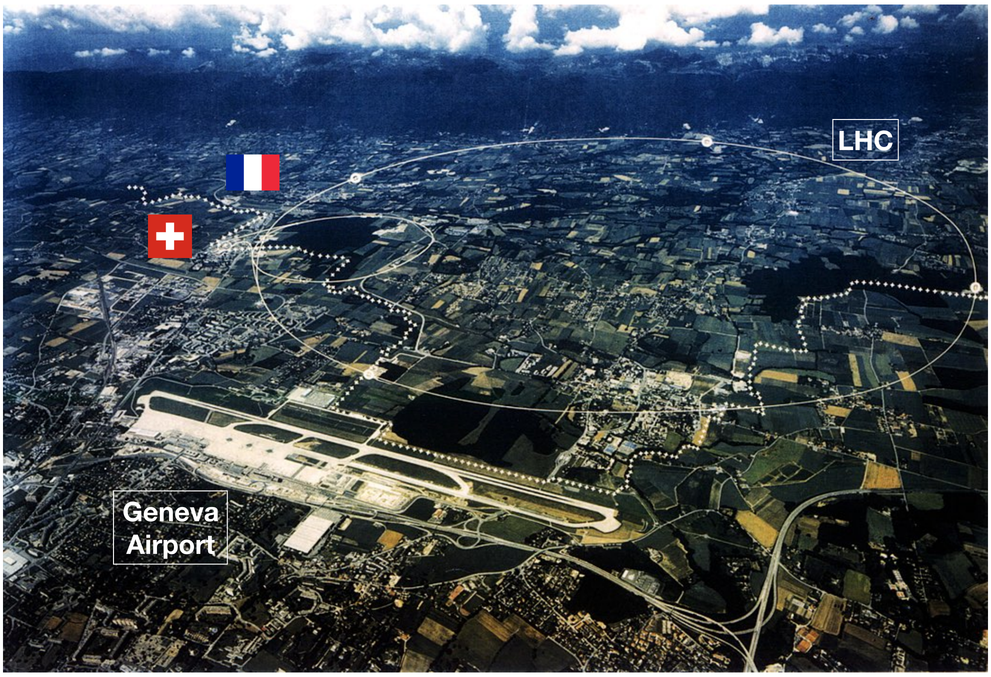

Figure 2: Aerial view of the location of the LHC. An approximation of the location of the circular underground tunnel is marked on the picture.

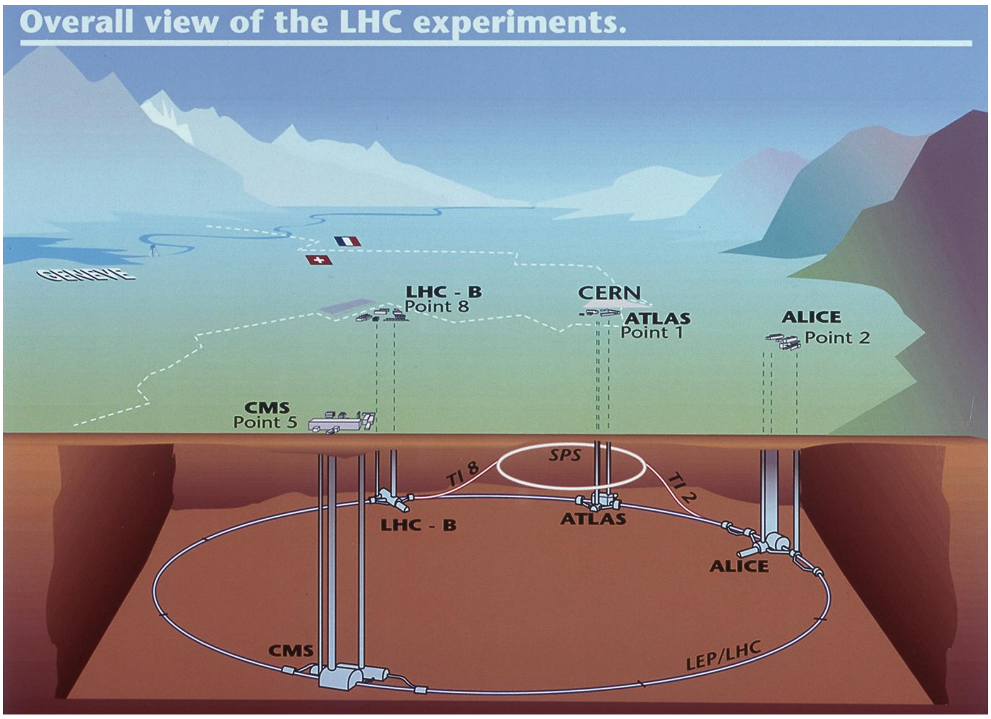

Figure 3: Location of the LHC and its four main experiments.

2.2 ATLAS and CMS

ATLAS [4] and CMS [5] are general-purpose particle detectors which are built in a cylindrical shape around the LHC beam pipe, in a way that the interaction point is at the centre of the detector. They consist of several layers which can be categorised into four main parts: the tracker, the electromagnetic calorimeter, the hadronic calorimeter and the muon spectrometer.

The tracker is the most inner part in these detectors. It detects the trajectory of electrically charged particles. When charged particles travel through the tracker, their interactions with the material leave behind hits which then are used to reconstruct a track. The magnetic field surrounding the tracker causes the particle trajectory to curve. The direction of the curve reveals the charge sign of the particle and the degree of curvature reveals the momentum of the particle.

The tracker is surrounded by the electromagnetic calorimeter. It is designed to measure the energies of the electrons and the photons. The electrons and photons interact with the material in electromagnetic calorimeter and deposit all their energy and are absorbed.

The next layer is the hadronic calorimeter, designed to measure the energy of the hadrons. Hadrons, which can pass through the electromagnetic calorimeter loosing only a fraction of their energy, interact with the specific energy-absorbing material and deposit all their energy and stop.

Muons pass through all the tracker system and are not absorbed by any of the calorimeters. In fact, the only known particles that penetrate beyond the hadron calorimeter layer are muons and neutrinos. So the most outer layer of the detector is specifically designed for the muon detection. The muon system identifies muons and measure their momentum in combination with the information from the tracking detectors.

The footprints of different particles in the CMS detector are schematically shown in Figure 4, and for the ATLAS detector in Figure 5. The dashed lines indicate the trajectory of the particles passing undetected. For example, as photons are electrically neutral, they are not detected by the tracker, but when they enter the electromagnetic calorimeter they interact with its material and produces an electromagnetic shower. Hadrons form a hadronic shower in the hadronic calorimeter, and before that if they are charged hadrons (like proton or pion) their track is seen by the tracker. Muon tracks are detected by the tracker, as they are charged particles, but they do not produce any shower when they travel through the calorimeters. They enter the Muon spectrometer system where once again their tracks are seen and then they leave the detector. Neutrinos are neutral and interact very weakly with the matter, therefore they go through the detector and leave it without being seen by the detector. But since a neutrino carries a fraction of the initial energy of the collision, they can be indirectly detected as missing

[4] **A** **T**oroidal **L**HC **A**pparatu**S**

[5] **C**ompact **M**uon **S**olenoid

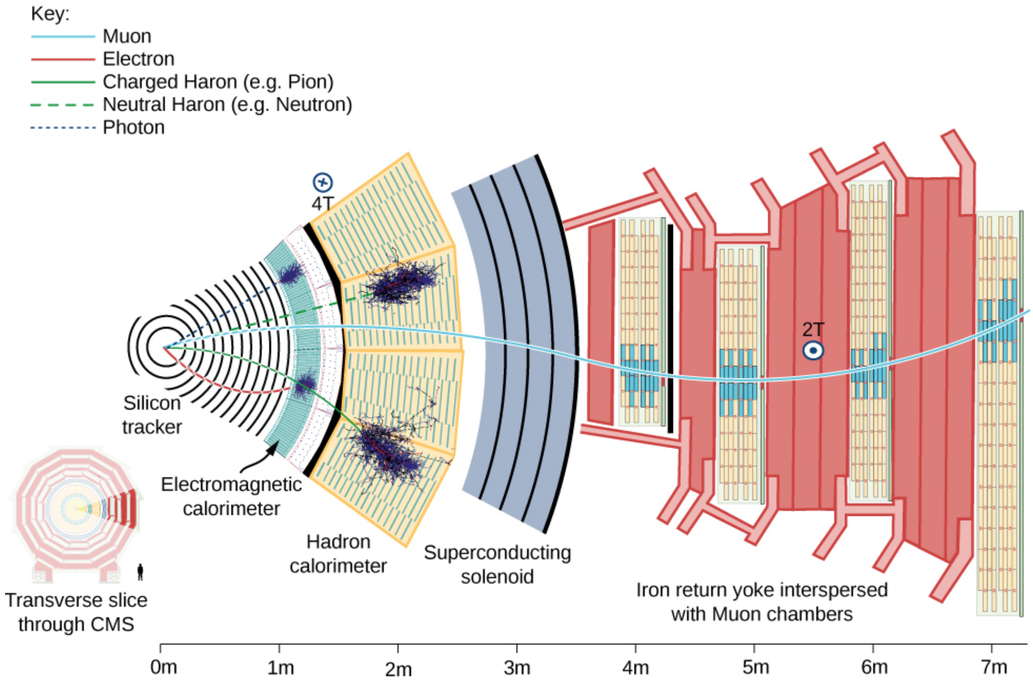

Figure 4: A computer-generated image representing how the CMS detector works.

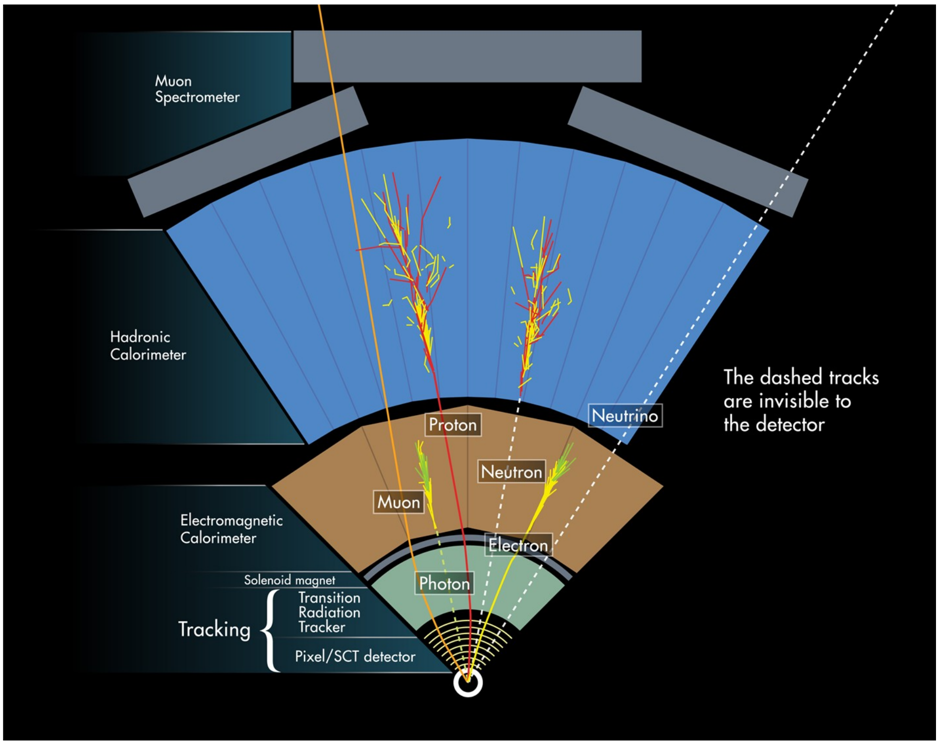

Figure 5: A computer-generated image representing how the ATLAS detector works.

energy in the transverse plane owing to energy conservation.

Informative animations can be found in the ATLAS and CMS Youtube channels, which help to understand how these two detectors work:
https://www.youtube.com/TheATLASExperiment
https://www.youtube.com/CMSExperimentTV

3 Top quarks

The existence of the top quark was postulated first in 1973, and it finally was discovered in 1995 by the CDF and D0 experiments at the Tevatron collider at Fermilab. Today even after many years passed from its discovery, top quark remains an important topic of intense research.

Top quark is by far the heaviest known elementary particle (see Figure 1). It has a very short lifetime and it decays before it can form a hadron with other quarks. This makes the top quark unique among all the quarks.

At hadron colliders, top quarks are produced dominantly in pairs of top-antitop quarks ($t\bar{t}$):

$$q\bar{q} \to t\bar{t}$$

$$gg \to t\bar{t}$$

Top quark decays into a W boson and a b quark with a branching ratio of almost 100%. The W boson is also not stable and further decays either into a lepton and a neutrino or into two quarks:

$$t \to bW^+,\ \bar{t} \to \bar{b}W^-$$

$$W^+ \to l^+\nu_l\ ,\ W^- \to l^-\bar{\nu}_l\ (l = e, \mu, \tau) \text{ or } W^\pm \to q\bar{q}'$$

Therefore, three $t\bar{t}$ final states can be classified according to the decay modes of the W bosons:

- The dilepton channel, where both W bosons decay leptonicaly (i.e. into lepton and neutrino),
- The semileptonic channel, also known as single-lepton channel, where one W bosons decays leptonicaly and the other one hadronicaly (i.e. into quarks),
- The all-hadronic channel, where both W bosons decay hadronicaly.

A $t\bar{t}$ pair produced in a hadron collider is identified through the detection of the b quarks and the decay products of the W bosons. The neutrino(s) decayed from the W boson(s) can not be seen by the detector and is(are) manifested as missing energy. The b quarks and the quarks which are decayed from the W boson(s) form jets, as was explained before in Section 1.6. In addition, the radiation of gluons either in the initial or final state are possible, which results in the formation of additional jets. Hence, the signature of the three mentioned final states in the detector are:

$$t\bar{t} \rightarrow b\bar{b}W^+W^- \rightarrow \left| \begin{array}{c} \text{2 leptons, missing energy, 2 jets (+ extra jets)} \\ \text{1 lepton, missing energy, 4 jets (+ extra jets)} \\ \text{6 jets (+ extra jets)} \end{array} \right.$$

In this lab course we only focus on the semi-leptonic channel. Figure 6 shows an illustration of the $t\bar{t}$ pair production in the semi-leptonic final state.

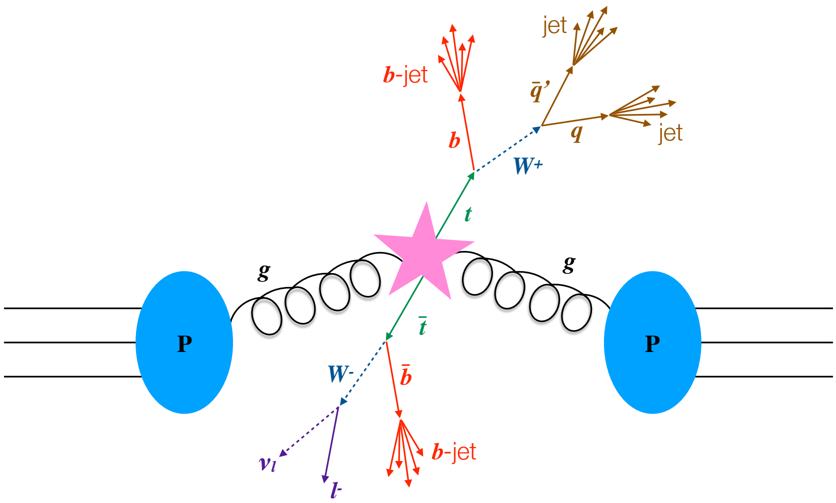

Figure 6: Ilustration of an example of the production and decay of a top-antitop quark pair in a proton-proton collision. Here, the $t\bar{t}$ pair is produced from incoming gluons, and the decay channel is the semileptonic channel.

4 The experiment

4.1 Analysis framework

4.1.1 ROOT

ROOT is a modular scientific software framework. It provides all the functionalities needed to deal with big data processing, statistical analysis, visualisation and storage. It is mainly written in C++ but integrated with other languages such as Python and R[6].

ROOT is already installed and setup on the computer that you will use for this lab course. You will use the HEPTutorial package for the data analysis

[6]Retrieved from https://root.cern.ch

and the components of this package inherit from classes of ROOT. Also, the data and simulation samples that you will use are in ROOT trees format.

4.1.2 HEPTutorial package

The package HEPTutorial is an example framework which will be used to do the data analysis for this lab course. The infrastructure, including reading in the data files, and example codes are ready for you. You just need to add some lines of code for each assignment into the already available codes, as will be described in the assignments.

The package is available on the computer that you will use for this lab course. First, change to the directory that contains the package, by typing in at the linux command line:

```
cd /home/TopExperiment/HEPTutorial
```

Then, compile the package by typing in at the linux command line:

```
make
```

This creates an executable named `example.x`. You can now simply run the analysis by typing the command:

```
./example.x
```

This will create two output file, results.pdf and results-MC.pdf, which contain all the plots from the analysis. You will also see result of counting number of events as print out on the Terminal where you executed the program.

For more advanced students, explanations for individual components of the package are given in Appendix B.

4.2 Data and simulation samples

The data and simulation samples are stored as ROOT trees format. The tree contains a collection of variables which are filled once per event. They are listed in Appendix A, with their data type and further explanations.

4.2.1 "*Truth*" vs "*Reconstructed*"

For some part of the assignments you will use only the simulation samples and for some parts you will use the real data and compare it with the simulated samples. The simulated samples consists on events that are randomly generated according to the production cross section of the different processes that can occur in the proton-proton collisions. The events are generated in the full phase space. The information of the generated particle's four momentum and their decays products is stored. This is referred to as *truth* information. In a second step the passing of the particles (the decay products of the hard interaction) throught the detector is simulated (*reconstruction level*).

This means that with a simulation sample, you have access to *truth* information, meaning that if our detector was ideal and would have detected and recorded all the events exactly as they happened. Another category of information is called *reconstructed* which corresponds to what the detector, which in reality is not ideal, actually is able to detect and record.

One can then know about the efficiency of the detector by comparing the *truth* with *reconstructed* information. The *truth* information is only available in the simulation, and not in the real data, because we have generated the simulated events ourselves so we know what has happened in the generation level, before any detector effect were added to it. But for the real data, obviously we only have the information seen and recorded by the detector, therefore only the *reconstructed* information.

In the case that one wants to compare the real data with the simulated samples in order to find out how much of the observed data is expected to be signal and how much of it is background, the *reconstructed* information of the simulated samples should be used.

4.3 Assignments

4.3.1 Assignment 1: Branching ratio

Using the truth information from the $t\bar{t}$ simulation sample, find out how often a $t\bar{t}$ pair in its semi-leptonic decay channel decays to an electron, or a muon, or a tau lepton.

Open the code file `MyAnalysis.C` with your favourite editor program and go to the line below, to see the hints:

```
// Assignment 1 :  Calculation of branching ratio
```

4.3.2 Assignment 2: Kinematic distributions $t\bar{t}$ decay products

Using the truth information from the $t\bar{t}$ simulation sample, plot some kinematics distributions of the $t\bar{t}$ decay products, like p_{T} distribution of the leptons (the exact kinematics variables that you should plot will be assigned to you by your tutor).

In `MyAnalysis.C`, go to the line below for hints:

```
// Assignment 2 :  Kinematic variables of "Generated" particles
```

4.3.3 Assignment 3: W boson mass

Using the truth information from the $t\bar{t}$ simulation sample, construct the W boson mass from its decay products.

For this you need to calculate the invariant mass of the lepton and the neutrino decayed from the W boson, and the invariant mass of the two quarks that are decayed from the W boson. This information is available in truth level, and you can find from the list of variables stored in data files (Appendix A) the proper variable which you should use to construct the four-vector of these decay products. Then you can use the function `M()` to get the invariant mass.

In `MyAnalysis.C`, go to the line below for hints:

```
// Assignment 3 :  W boson mass
```

4.3.4 Assignment 4: top quark mass at truth level

Using the truth information from the $t\bar{t}$ simulation sample, construct the top quark mass from its decay products. This assignment can be done similar as to the previous assignment. In addition, plot the p_{T} distribution of the two top quarks.

In `MyAnalysis.C`, go to the line below for hints:

```
// Assignment 4 :  top quark mass
```

4.3.5 Assignment 5: Cross section measurement of $t\bar{t}$

For this assignment, you need to select the $t\bar{t}$ events (referred to as *signal events*) from all the recorded data events. You already know from previous pages what is the signature of a $t\bar{t}$ event, so you can decide if you want to add, change or remove any of the selection cuts in this part of the code.

Not all events in data that survive the $t\bar{t}$ event selection criteria are actually signal. No matter how the event selection cuts are carefully chosen, one should always consider the probability that some background events contaminate because they have a similar signature as the signal. In this experiment we estimate the number of background events with the simulation samples. All the other simulated events that you see in the plots which are not $t\bar{t}$, are the backgrounds. You can count their numbers and subtract the total background from the observed number of events in real data:

$$N_{observed\ signal} = N_{data} - N_{Bkg}$$

From equation 1, you have:

$$\sigma = \frac{(N_{data} - N_{Bkg})}{A.C.L} \tag{2}$$

The acceptance factor, A, can be calculated with the help of the truth information in $t\bar{t}$ simulation sample. It is the fraction of the generated $t\bar{t}$ events that fall into fiducial region, to the total generated $t\bar{t}$ events available in the simulation sample:

$$A = \frac{N_{truth}^{fiducial}}{N_{truth}^{all}} \tag{3}$$

You can get $N_{truth}^{fiducial}$ by counting the number of TRUE events in $t\bar{t}$ simulation sample that survive the selection cuts (hence events that fall into our measurement fiducial region).

Efficiency factor, C, can be calculated by comparison of the truth and reconstructed information in $t\bar{t}$ simulation sample, as following:

$$C = \frac{N_{reconstructed}^{fiducial}}{N_{truth}^{fiducial}} \tag{4}$$

You can get $N_{reconstructed}^{fiducial}$ by counting number of RECONSTRUCTED events in $t\bar{t}$ simulation sample that survive the selection cuts (hence events that fall into our measurement fiducial region and are seen by the dector).

The dataset that we use in this lab course has a luminosity of L = 50 pb^{-1}.

With this information, you can now calculate the $t\bar{t}$ cross section. Do not forget to report the uncertainty of the cross section. For simplicity, it is enough to only take into account the statistical uncertainty of the $N_{observed\ signal}$ when calculating the cross section uncertainty.

In `MyAnalysis.C`, go to the line below for hints:

```
// Assignment 5:  Cross section measurement of ttbar
```

4.3.6 Assignment 6: W boson and top quark reconstruction

The aim of this assignment is to reconstruct the masses of the W boson and the top quark that decay hadronically (at reconstruction level). Since the W boson and the top quarks are not observed in the detector, we need to reconstruct their 4-momentum from their decay products. For this task one should use only the events that pass the full selection in data and in simulation, to have a very clean sample of $t\bar{t}$ events.

The first step is to identify which jets most likely correspond to the W boson and calculate their invariant mass. How does the distribution compare to the distribution at truth level in Assignment 3?

The following step is to reconstruct the hadronically decaying top quark from the sum of three jets, two from the W boson and one additional b-jet. Since there is more than one b-jet, there are several combinations. Try to compute the most likely one. Compare these distributions at reconstrution level with the one at truth level in Assignment 4. Which could be the reasons for the differences, if any?

In `MyAnalysis.C`, go to the line below for hints:

```
// Assignment 6:  W boson and top quark reconstruction
```

Appendix A: Data and simulation samples

The data and MC simulated events are stored in ROOT trees format. The tree contains a collection of variables which are filled once per event. They are listed in the following, together with their data type and additional explanations.

List of data files

- **Real data:** data.root
- **Simulation sample for signal:** ttbar.root
- **Simulation samples for backgrounds:** dy.root, qcd.root, single_top.root, wjets.root, ww.root, wz.root, zz.root

List of variables[7]

- `NJet` (integer): number of jets in the event.
- `Jet_Px[NJet]` (float): x-component of jet momentum. This is an array of size NJet, where a maximum of twenty jets are stored (NJet < 21). If there are more than twenty jets in the event, only the twenty most energetic are stored. Only jets with $p_T > 30$ GeV are stored.
- `Jet_Py[NJet]` (float): y-component of jet momentum, otherwise same as `Jet_Px[NJet]`.
- `Jet_Pz[NJet]` (float): z-component of jet momentum, otherwise same as `Jet_Px[NJet]`.
- `Jet_E[NJet]` (float): energy of the jet, otherwise same as `Jet_Px[NJet]`. Note that the four components `Jet_Px`, `Jet_Py`, `Jet_Pz` and `et_E` constitute a four-vector which fully describes the kinematics of a jet.
- `Jet_btag[NJet]` (float): b-tagging discriminator. This quantity is obtained from an algorithm that identifies B-hadron decays within a jet. It is correlated with the lifetime of the B-hadron. Higher values indicate a higher probability that the jet originates from a b-quark. Important: The discriminator has small performance differences in data and simulation. To account for this, simulated events have to be reweighted by a factor of $\sim$0.9 per required b-tagged quark.
- `Jet_ID[NJet]` (bool): Jet quality identifier to distiguish between real jets (induced by hadronic interactions) and detector noise. A good jet has true as value.

[7] Retrieved from: C. Sander, A. Schmidt, *CMS data analysis tutorial: Documentation and explanations*, September 2014

- `NMuon` (integer): number of muons in the event.

- `Muon_Px[NMuon]` (float): x-component of muon momentum. This is an array of size NMuon, where a maximum of five muons are stored (NMuon < 5). If there are more than five muons in the event, only the five most energetic are stored.

- `Muon_Py[NMuon]` (float): y-component of muon momentum, otherwise same as `Muon_Px[NMuon]`.

- `Muon_Pz[NMuon]` (float): z-component of muon momentum, otherwise same as `Muon_Px[NMuon]`.

- `Muon_E[NMuon]` (float): energy of the muon, otherwise same as `Muon_Px[NMuon]`. Note that the four components `Muon_Px`, `Muon_Py`, `Muon_Pz` and `Muon_E` constitute a fourvector which fully describes the kinematics of a muon.

- `Muon_Charge[NMuon]` (integer): charge of the muon. It is determined from the curvature in the magnetic field and has values +1 or -1.

- `Muon_Iso[NMuon]` (float): muon isolation. This variable is a measure for the amount of detector activity around that muon. Muons within jets are accompanied by close-by tracks and deposits in the calorimeters, leading to a large values of `Muon_Iso`. On the other hand, muons from W bosons are isolated and have small values of `Muon_Iso`.

- `NElectron` (integer): same as for muons above, but for electrons.

- `Electron_Px[NElectron]` (float): same as for muons above, but for electrons.

- `Electron_Py[NElectron]` (float): same as for muons above, but for electrons.

- `Electron_Pz[NElectron]` (float): same as for muons above, but for electrons.

- `Electron_E[NElectron]` (float): same as for muons above, but for electrons.

- `Electron_Charge[NElectron]` (integer): same as for muons above, but for electrons.

- `Electron_Iso[NElectron]` (float): same as for muons above, but for electrons.

- `NPhoton` (integer): same as for muons above, but for photons.

- `Phtoton_Px[NPhoton]` (float): same as for muons above, but for photons.

- `Photon_Py[NPhoton]` (float): same as for muons above, but for photons.
- `Photon_Pz[NPhoton]` (float): same as for muons above, but for photons.
- `Photon_E[NPhoton]` (float): same as for muons above, but for photons.
- `Photon_Iso[NPhoton]` (float): same as for muons above, but for photons.
- `MET_px` (float): x-component of the missing energy. Due to the hermetic coverage of the LHC detectors and the negligible transverse boost of the initial state, the transverse momentum sum of all detector objects (jets, muons, etc...) must be zero. This is required by energy and momentum conservation. Objects which escape the detector, such as neutrinos, are causing a "missing" transverse energy which can be measured and associated to the neutrino.
- `MET_py` (float): y-component of the missing energy.
- `NPrimaryVertices` (integer): the number of proton-proton interaction vertices. Due to the high LHC luminosity several protons within one bunch crossing can collide. This is usually referred to as "pileup". The spread of these vertices is several centimeters in longitudinal direction and only micrometers in the transverse direction.
- `triggerIsoMu24` (bool): the trigger bit. It is "true" if the event is triggerd and "false" if the event is not triggered (data can only contain triggered events).

All quantities discussed above are actually measured by the detector, and are referred to as reconstructed, as mentioned before. They are available both in simulation and real data. The following variables are ONLY available in simulation and are referred to as truth, as mentioned before. They refer to the true $t\bar{t}$ decay cascade on generator level.

Please note that these variables are only filled for semi-leptonic $t\bar{t}$ decays. They are set to zero for all-hadronic and di-leptonic decays. They are also zero in the background simulation samples.

- `MChadronicBottom_px` (float): x-compoment of the b-quark from the top decay belonging to the hadronic branch.
- `MChadronicBottom_py` (float): y-compoment ...
- `MChadronicBottom_pz` (float): z-compoment ...
- `MChadronicWDecayQuark_px` (float): x-component of the quark from the hadronic W boson decay

- `MChadronicWDecayQuark_py` (float): y-component ...
- `MChadronicWDecayQuark_pz` (float): z-component ...
- `MChadronicWDecayQuarkBar_px` (float): x-component of the anti-quark from the hadronic W boson decay
- `MChadronicWDecayQuarkBar_py` (float): y-component ...
- `MChadronicWDecayQuarkBar_pz` (float): z-component ...
- `MCleptonicBottom_px` (float): x-compoment of the b-quark from the top decay belonging to the leptonic branch.
- `MCleptonicBottom_py` (float): y-compoment ...
- `MCleptonicBottom_pz` (float): z-compoment ...
- `MClepton_px`] (float): x-component of the lepton (electron, muon, tau) from the leptonic W boson decay.
- `MClepton_py` (float): y-component ...
- `MClepton_pz` (float): z-component ...
- `MCleptonPDGid` (integer): particle "ID" of the lepton. Possible values are 11 for electrons, 13 for muons, 15 for taus. Negative numbers indicate anti-particles.
- `MCneutrino_px` (float): x-component of the neutrino from the leptonic W boson decay.
- `MCneutrino_py` (float): y-component ...
- `MCneutrino_pz` (float): z-component ...
- `EventWeight` (float): weight factor to be applied to simulated events due to different sample sizes.

Appendix B: More detail on Analysis framework

A description for the individual components of the HEPTutorial package are given here[8].

- `example.C`: This is the first starting point. It contains the `main()` function which is necessary for any C++ program. The first step is to create instances of `MyAnalysis` which is implemented in the files `MyAnalysis.h` and `MyAnalysis.C` (explained in the next item). The `TChain` represents the ROOT tree discussed in Appendix A. The files which should be read from disk are specified in the function `Add(filename)`. The tree is then read and processed by the `Process()` function which takes a `MyAnalyis` as argument. The real work is then done in the `Process()` function of the `MyAnalysis` class which is discussed in the next two items. You should instantiate a separate `MyAnalysis` class for each data or MC sample that you want to process. The end of the `main()` function shows how to access and write out the histograms which are filled inside the `MyAnalysis` class.

- `MyAnalysis.h`: Definition of the class `MyAnalysis`, which inherits from `TSelector` (see the ROOT reference manual for details about `TSelector`). The constructor takes a global scaling factor (by default 1.0) which is applied to all events in the given sample (i.e. multiplied to the individual event weights). This global scaling factor can be used to normalise the MC cross sections. `MyAnalysis` contains the declaration of all variables which are contained in the ROOT tree (see Appendix A). It also declares variables and functions that will be used in your analysis, such as histogram pointers (type `TH1F*`). Some containers of type `vector<...>` are also declared here. These containers will hold the helper classes representing jets (`MyJet`) or muons (`MyMuon`). Finally, the `MyAnalysis::Init()` function makes the connection between the ROOT tree (stored on disk) and the variables which are kept in memory.

- `MyAnalysis.C`: The two main functions which are called automatically while processing the ROOT trees are `SlaveBegin()` and `Process()`. The `SlaveBegin()` function is called only once per job, just before the processing of the events start. You see that it is used for booking histograms (assigning the histograms to the pointers defined in the `MyAnalysis.h` file). The `Process()` function is called automatically for every single event. This is the place where the core of the analysis happens. In the existing example `Process()` calls a subroutine

[8] Retrieved from: C. Sander, A. Schmidt, *CMS data analysis tutorial: Documentation and explanations*, September 2014

`BuildEvent()` which takes care of filling some of the kinematic variables into convenient representations as `MyJet` or `MyMuon`, which are essentially four-vectors (explained in the next item). After building the event, a very simple example analysis is performed in `Process()` which fills the transverse momentum pt of a muon into a histogram.

- `MyMuon.h/MyMuon.C`: Inherits from `TLorentzVector` which is basically a four-vector (see the ROOT reference guide for details about `TLorentzVector`). It adds additional information about the muon charge and muon isolation variables to the four-vector.

- `MyElectron.h/MyElectron.C`: Same as `MyMuon` but for electrons.

- `MyJet.h/MyJet.C`: Inherits from `TLorentzVector`, adds additional information about the b-tagging variable to the four-vector. The b-tagging variable is correlated to the probability that the jet originates from a b-quark. Large values of this variable means high probability for a b-quark jet. The function `IsBTagged()` applies a cut to the b-tag discriminator and returns a boolean decision (true or false). The function `GetJetID()` returns true if a jet fulfils basic quality criteria, else it returns false.

- `Plotter.h/Plotter.C`: A tool which can be used for automatic plotting of a set of histograms which are stored in a `std::vector`. Please see `example.C` on how to use it.