Vrije Universiteit Brussel

FACULTEIT WETENSCHAPPEN EN BIO-INGENIEURSWETENSCHAPPEN

Anomalous Couplings Project

Progress Report and documentation of Tools

Annik Olbrechts

Started 4 April 2014
Version: January 20, 2015

Contents

Chapter 1

Pending issues

- Does this created FeynRules model still contain Effective Field Theory?
- If the kinematics doesn't change for coupling parameters larger than 1, how is it than possible to differentiate the different configurations which will be studied?
- Understand *CorrectPhi* comment in MadWeight name
  $\rightarrow$ Maybe related with phi issues of neutrino's ...
- EventWeight calculated in analyzer should be integrated into MadWeight output (Should MadWeight weight just be multiplied with the EventWeight for each event?) **Otherwise all the effort to include JES and PU has no influence at all.**
- Update EventNumberInformation file to only cout the information when the event has passed the eventSelection requirements. Otherwise just discard this event from the output file in order to avoid a long .txt file and long running time ...
- Construct nTuple in such a way that the likelihood and weight distributions can be calculated for specific p_T cuts without having to run the entire base code again. So perhaps also necessary to include some of the different python scripts which have been created. Hence avoid spending too much time on improving them.
- Need to understand how MadWeight deals with the permutations ... Should this permutation of the light jets be done within MadWeight or should two .lhco files be created and sent separately to MadWeight. If the latter is the case, how should they be combined afterwards ?
- Specific MadWeight question for the latest version:
  The 'refine' option allows you to realunch the computation of the weights which have a precision lower than X. ***But how can you find this precision?***
- Should a ScaleFactor correction be applied for the Branching ratio ?
  This is not the case in code of James, but should find recommendations somewhere.
- **To Check:** Does MadWeight give a different weight when the Neutrino Mass changes?
  $\rightarrow$ Currently mass is manually set to zero ...
- Check whether the created root file can be trusted when the number of considered b-tags is greater than 1 ... *Why this comment??*

Chapter 2

Used Tools and Techniques

2.1 FeynRules

Should be understood why the mass of the top quark is equal to 180 within the created FeynRules model!

2.2 MadGraph

In the beginning of this analysis only the MadGraph version $MadGraph\_v155$ was used since this was the only one compatible with the FeynRules version used to create the model. According to a mail of Olivier Mattelaere on 6 February 2014 the newer version is currently compatible as well.
The newer MadGraph version MadGraph5_aMC@NLO was released in the spring of 2014, together with a new MadWeight version. It is expected that the calculated cross sections are identical between these two MadGraph versions and this has been examined. *Put some results here!*

In this project MadGraph is only used to create .lhe files and study the variation of the cross section in order to understand the created model. The following two python scripts have been created for this reason:

$/user/aolbrech/AnomalousCouplings/MadGraph\_v155/MassiveLeptons/$
$MadGraph5\_v1\_5\_5/Wtb\_ttbarSemiElMinus/RVL\_RVR\_XSGrid003.py$
$/user/aolbrech/AnomalousCouplings/MadGraph\_v155/MassiveLeptons/$
$MadGraph5\_v1\_5\_5/XSScript.py$

The first script automatically creates all the desired configurations for different RVL, RVR, Lepton Pt Cut and Jet Pt Cut values and stores them in the desired output directory. In order to save some disk space the events.lhe file is deleted directly and only the unweighted_events.lhe files are kept for analysis. *According to Alexis these are the only relevant ones so understand what is the difference between the two....*
The second script loops over all the different directories and creates a .txt table and a .pdf table with the cross section results.

In order to use MadGraph and calculate the cross section values for the desired processes and decay channels the following commands should be used:

$$./bin/mg5_aMC \quad (./bin/mg5 \quad \text{for old MW version}) \tag{2.1}$$
$$import \quad model \quad MassiveLeptons \tag{2.2}$$
$$\begin{aligned} generate\ pp > tt \sim, (t > bw+, w+ > mu + vm) \\ , (t \sim > b \sim w-, w- > jj)@1QED = 2 \end{aligned} \tag{2.3}$$
$$outputdirName \tag{2.4}$$
$$exit \tag{2.5}$$

The *index.html* file in this new directory contains all the Feynmann Diagrams which correspond to the generated process and can be opened using $firefox$ on mtop.
The FeynRules model is created in such a way that it contains a **NP**, **QCD**, **QED** and **TEST** variable representing the different interaction vertices. Hence asking **QED** = 2 results in the desired 16 independent diagrams, namely an altered top quark decay while the other interaction vertices are still described by the Standard Model.

Whenever the cross section should be calculated for a specific process the following configuration files in the Cards directory should be updated:

- **run_card.dat** where the number of generated events is defined together with the beam energy of the considered collision process. This file also contains all kinematic cuts which can be applied. Since these will influence generator events they will have a different effect then the jet-level cuts applied in the event selection.

- **param_card.dat** where all the parameters of the considered model configuration can be be defined.

- **proc_card.dat** describes the type of generated event and the imported models. This file shouldn't be altered but is particularly useful to ensure that the correct event topology is considered.

Each of these files can still be adapted after the *./bin/generate_events* command since the MadGraph software always asks whether one or more of the configuration files should be changed before the command is executed (with a waiting time of 60s).
As a consequence, when using a script to run MadGraph continously, the **me5_configuration.txt** file should be adapted to enable this waiting time. Otherwise using nohup to run the script results in a crash and the termination of the script.

2.3 MadWeight

As was the case with MadGraph, two different versions of the MadWeight event generator exist. All original tests have been performed with the older one, installed under *madweight_mc_perm*, and used until July 2014. The newer one, installed as $MadGraph5\_aMC@NLO$, is supposed to be less CPU intensive and allows the user to split the interaction points in two distinct steps. More detail about the different options can be found in 2.3.1. Since September 2014 this newest version is used as standard, but both versions are still installed on localgrid and the user disks.

Each version can be installed using the following command:

- bzr[1] branch lp:∼maddevelopers/madgraph5/madweight

- bzr branch lp:∼maddevelopers/mg5amcnlo/madweight

After a fresh MadWeight install with the *bzr* command the created MassiveLeptons FeynRules model should be copied to the "model" directory. Otherwise it cannot be accessed by the MadWeight event generator.
In MadWeight each specific model can be generated but will be stored in a separate directory. Hence special care should be awarded to the chosen directory name. The necessary commands to create such a new MadWeight directory are listed here:

1. ./bin/mg5_aMC (The mg5 executable should not be used!)

2. import model MassiveLeptons

3. generate p p > t t∼ , (t > b w+ , w+ > mu+ vm) , (t∼ > b∼ w- , w- > j j) @1 QED = 2

4. output madweight **Name**

5. exit

In order to actually calculate Matrix Elements using this MadWeight software the relevant configuration files should be adapted in order to select the correct configuration. Just as was the case when using the MadGraph software, the concerned files are the **run_card.dat**, **param_card.dat** and the **proc_card.dat**.
The only relevant new card is the **MadWeight_card.dat** file where the simulated parameters of the corresponding **param_card.dat** can be introduced. For the rest this file contains all the different run options which can be set and which will be discussed in detail in Section 2.3.1.

As a next step the correct Transfer Function should be chosen from the list of available ones. This list is given after the following two commands have been executed and allows the use of tab-completetion. If no option is chosen within the 60s waiting time, the underlined one is chosen.

- ./bin/mw_options

- define_transfer_fct

Finally MadWeight is completely initialized and can be used. Running MadWeight is done with the *./bin/madweight* command combined with the option −1 to create all the param_cards, option −2 to ..., option −3 to ..., option −4 to ..., option −6 for actually starting to run MadWeight and afterwards with option −8 to collect all the weights and store them in the corresponding Events directory.

[1] The bzr package has been installed on the m-machines, but not on mtop. Due to some conflicht with the *libz.so* package it cannot be executed after initializing the alias *setMGpython*. More information about this issue can be found in 2.3.3

2.3.1 Synchronizing MadWeight with localgrid submission

Running the MadWeight event generator is preferable done with localgrid submission since this allows to split the *.lhco* file containing all event in multiple jobs. However the standard MadWeight configuration is not directly compatible with the IIHE cluster submission. For this to work out-of-the-box the following adaptations should be done: For this first step the following two files have to be changed in the /blablaMadWeight/bin/internal directory:

- madweight_interface.py
- cluster.py

Once jobs are running on localgrid their performance and running time can be checked easily and, if needed, they can be killed. This should be avoided as much as possible since it can influence the priority of the following jobs sent.

$$qstat \ \ @cream02 \,|\, grep \ \ aolbrech \tag{2.6}$$
$$qdel \ \ 394402.cream02 \tag{2.7}$$

As mentioned before, the **MadWeight_card.dat** contains a bunch of run options which can be defined. One of these options is the number of events and the number of events in one job. Since the number of simultaneous jobs on localgrid is limited to 2000 the number of processed events in each job should be taken as low as possible without exceeding this limit. It should be avoided to run MadWeight with a large number of events in one single job since this will result in a very long walltime and reduced priority.

Another important factor for the MadWeight event generator is defining the number of interaction points which have to be considered. This number has to be rather large (around 10 000 - 30 000 according to Lieselotte **TO CHECK MYSELF**) to ensure the uncertainty on the weight to be smaller than the weight itself **Is this really a consequence?**.

In the newest MadWeight version, an option to split the interaction in two distinct steps is added. **What is ideal configuration?**

Adapting MadWeight to run continously on localgrid A single test with the following configuration (5000 events - 20 events/job - 10 000 initPoints) performed with the oldest MadWeight version[2] took almost 16h to finish. This implies that running the full generator event sample (> 2 470 000 events) would take more than 329 days![3] Even the full reconstructed event sample (> 210 000 events) requires more than 28 days of running this way. It is expected that with the newest MadWeight version this time can be reduced significantly, but still a long running time should be foreseen.

Big improvements on running time can possible be reached if the MadWeight event generator is optimized for running as efficient as possible on localgrid. One characteristic of running MadWeight on localgrid which significantly influences the running time is the fact that it is not capable of continously submitting jobs. In stead the maximum allowed number of 2000 jobs is sent and one then has to wait until they are all finished before

[2]TODO: Compare needed time of running explicitely against the new MadWeight version!

[3]From this can be concluded that the events on generator level should never be all run. Only a limited selection of these events should be considered.

a new bunch can be submitted. It would be much more favourable if the MadWeight submission script would submit a new job as soon as one of the 2000 has finished.

According to Olivier Devroede this could indeed be possible but since this is a change in the MadWeight cluster configuration it has to be done by the MadWeight developers. It cannot be adapted by the grid admins.

2.3.2 Influence of the used Transfer Function

2.3.3 Complementarity with Git

In order to keep track of all the changes necessary for the configuration files of MadWeight, the localgrid MadWeight directory has been added to the AnomalousCouplings GitHub repository. This also allows to keep two separate branches for both the original E-dependent Transfer Functions and the newly created p_T-dependent Transfer Functions.
However this has an important restriction on the use of the m-machines for compiling and running MadWeight and for performing GitHub related activities. This restriction is caused by the initialization command needed to run MadWeight, namely activating python 2.7.3, which results in a conflict with the GitHub requirements. Therefore the GitHub related activities have to be done either in a separate terminal window where no python initialization is done or otherwise on *mtop* where this python command can't be executed. The compiling and running of MadWeight on the other hand has to be done in a different terminal window on the m-machines.
This implies that in the terminal window where the analysis is being executed no Git commands can be executed, like for example checking the active branch or changing to another branch. These commands are only allowed in the separate window.

2.4 MadAnalysis

At first sight it seems that since the update of the m-machines to Scientific Linux 6 ($slc6$) MadAnalysis can only be used on *mtop*. Whenever MadAnalysis is trying to be compiled on any of the other m-machines an error message about a missing library appears, while compiling on *mtop* works from the first try.

MadAnalysis is most useful in the expert mode since this allows to develop a personal analysis file where an event selection can be applied and specific kinematic variables, such as the $\cos\theta^*$ one, can be defined. Also for MadAnalysis two different versions exist, but only the expert mode in version $v112$ is compatible with the explanation in the manual ($arXiv : 1206.1599$). The expert mode of the more recent version, $v115$, does not work out-of-the-box and requires adaptations to the python files.

In order to start with a new analysis the following command has to be used:

$$./bin/ma5 \quad --expert \tag{2.8}$$

This results in a series of questions such as the name of the directory which has to be created and the name of the analysis. The latter one should not be taken too complex since it has to be entered each time a series of plots is created for this analysis. After the desired directory is created, the $Name$/SampleAnalyzer directory should be initialized by executing the following two commands:

$$source\ \ setup.sh \tag{2.9}$$
$$make \tag{2.10}$$

The actual analysis should be created in the Analysis directory, and a similar approach as in *user.cpp* and *user.h* should be adopted.
Everytime a change has been made to any these two files, *make* should be executed in the SampleAnalyzer directory in order to process the changes.

The different *.lhe* files[4] which should be considered should be defined in a *List.txt* files which is saved in the SampleAnalyzer directory.

The actual running of MadAnalysis is then done using the following command:

$$./SampleAnalyzer\ \ --analysis = "Name\ of\ analysis"\quad List.txt \tag{2.11}$$

2.4.1 Content of analysis file in MadAnalysis

The latest analysis file which has been used and containing all the necessary information can be found in the following directory on the m-machines:

$AnomalousCouplings/MadAnalysis\_v112/Wtb\_PtCutInfluence/SampleAnalyzer/Analysis$

The two analysis files with full detailed information are the $LeptonPtCutInfluence.cpp$ and the $JetPtCutInfluence.cpp$ files. They both consist of two different functions, namely the *Execute* and the $Finalize$ function. The first one allows to access the information of each event while the second one is only accessed once for each file. Therefore the particle content is reconstructed in the *Execute* function and the histograms for all the considered files are constructed in the $Finalize$ function. Currently these analyzer files looks at 28 different kinematic variables. All the kinematic information of each of the particles present in the expected semi-leptonic $t\bar{t}$ event is created.

In order to separate the two b-quarks in the event, the Particle Id of the leptonic top quark needs to be known. Therefore an integer $LeptonicTopPdgId$ is used and the kinematic information of the b-quarks can only be stored when this integer is different from zero. This will normally not result in by-passing the b-quark information since the events in the .lhe files are read in in the same order as they are created by the MadGraph command. So first the top quarks are considered and only then the final state particles.

In order to automatically create the 28 histograms for all the different files and distinguish the different RVR and RVL values, an automatic name-giving loop is developed. The name of the content of the histogram is each time combined with the correct name of the RVL/RVR content of the *.lhe* file and the decayChannel.

2.4.2 Analyzing the MadGraph files

Currenlty the created model should be completely understood and the behavior of the model when the coupling coefficients are larger than 1 should be investigated. Therefore

[4]MadAnalysis cannot process .lhe.gz files so they should be unpacked using *gunzip.lhe.gz*

new MadGraph files have been processed for the following configuration:

$$\begin{aligned} Re(V_L) &\in [0.7,\ 1.3] \\ Re(V_R) &\in [-0.3,\ 0.3] \end{aligned}$$

These files can be found in the following directory on the m-machines and contain 100 000 events.

$$/user/aolbrech/AnomalousCouplings/MadGraph\_v155/MassiveLeptons/$$
$$MadGraph5\_v1\_5\_5/Wtb\_ttbarSemiElMinus/ResultsXSGrid003$$

Chapter 3

Generator level bottlenecks

The goal is to obtain results on generator-level as fast as possible in order to ensure that no bottlenecks are encountered when running MadWeight. The advantage of only using these events is that one can be completely sure that they are actual semi-leptonic ttbar events. Hence MadWeight should not have any problems calculating the weight for these kind of events and no CPU time will be spent on uncorrect events.
This implies that any deviation from the expected results implies a bias, or even a problem, concerning the MadWeight output.

Once the results correspond to the expectations these preliminary results could be easily extended to reconstructed events. Finalizing the event selection then allows to fully trust the results obtained on reconstructed level and make sure that any new deviation can be explained by the influence of the applied event selection.
These results can then be used to optimize the event selection with respect to the MadWeight output and CPU time needed.

However the first results obtained with MadWeight in this personally FeynRules model did not result in the desired solution. It was observed that the minimum of the obtained $\ln(\mathcal{L})$ did not correspond the the expected Standard Model value of $(V_L, V_R)) = (1.0, 0.0)$. This was then investigated further and a possible solution was found in the missing cross section normalisation. Hence a detailed calculation of the cross section value of each point in the grid was performed, and also here some discrepancies were found. The most important one was the observation that the cross section values varied when moving from one MadGraph version to the other one. Hence also here a thourough investigation was performed.

3.1 Uncorrect $\ln(\mathcal{L})$ minimization

The first obtained MadWeight results for the enlarged grid ($V_L \in [0.8, 1.2]$ and $V_R \in [-1, 1]$) using only parton-level ttbar events did not result in the expected minimum of $(V_L, V_R) = (1, 0)$. This can be observed in Figure 3.1 which contains the distribution of the $\ln(\mathcal{L})$ for each point of the grid. Looking at these values in detail clearly shows that the minimum of this distribution can be found in both upper corners, and not in the expected center of the grid. It is even worse than at first sight, because the Standard Model value actually almost corresponds to the maximum of the distribution. The only two values which are still slightly higher are the $(0.9, 0.0)$ and $(0.8, 0.0)$ ones.

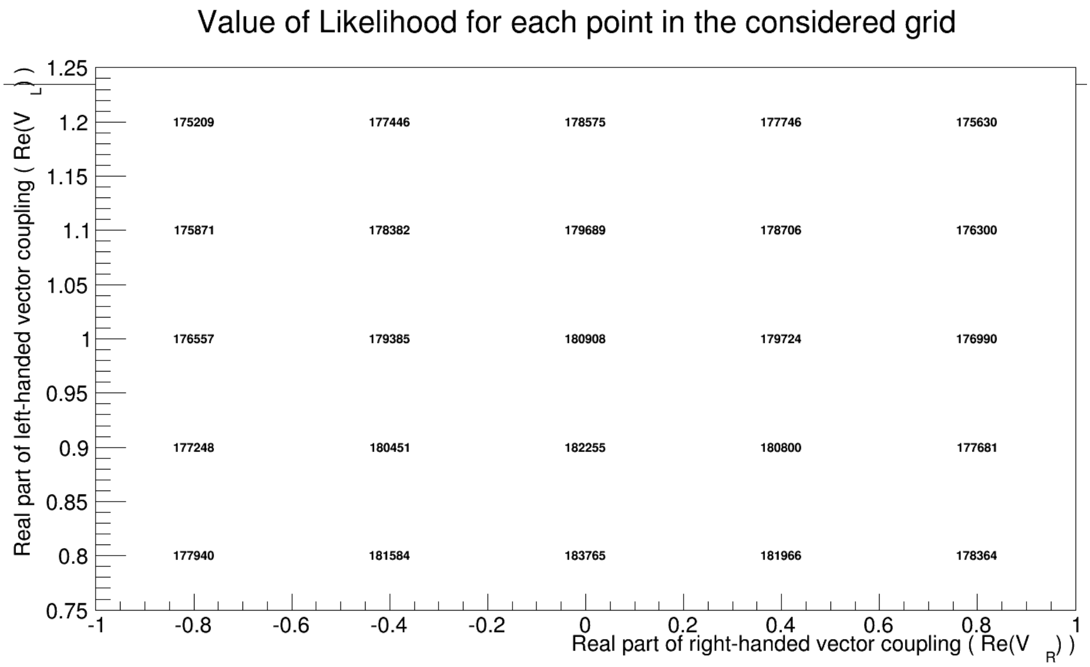


Figure 3.1: Distribution of the $\ln(\mathcal{L})$ for each point in the grid using 3200 parton-level positive semi-muonic ttbar events. The transfer function used to smear the parton-level kinematics is the single-gaussian function standard included in MadWeight.

The reason why the cross section normalisation is seen as a possible explanation can be understood from the left distribution in Figure 3.2. This shows the XS normalisation, $\frac{XS}{XS^{SM}}$, for each point in the grid. It clearly indicates the exactly opposite behavior as the $\ln(\mathcal{L})$ distribution which might hint at an uncorrected XS influence breaking down the $\ln(\mathcal{L})$ minimization.

This XS normalisation was originally applied to the Madweight output, but has been removed since the MadGraph5 version. However in the following papers, $arXiv$: 1101.2259 and $inspirehep$: 854451, motivation has been found for the original used implementation. The so-called general Matrix Element formula takes the following form:

$$P(y|a) = \frac{1}{\sigma(a) * Acc(a)} \int W(y|x,a) * Eff(x,a)|M(x,a)|^2 T(x,a)dx \tag{3.1}$$

$$\mathcal{L} = \prod P(y|a) \tag{3.2}$$

In this Equation, (3.1), the factor $\sigma(a)$ is defined the channel cross section and can actually be seen as a sort of XS normalisation factor. This factor was the one which was originally implemented in the older MadWeight versions but now is removed. However this still seems to suggest that the current normalisation $\frac{XS}{XS^{SM}}$ might be too simplistic and should be changed towards this channel cross section which is defined as:

$$\sigma(a) = \int_{X_i} |M(x,a)|^2 T(x,a)dx \tag{3.3}$$

Besides the correctness of the used XS normalisation, special care should be awarded to the implementation of this normalisation. At first it was uncorrectly multiplied with the $\ln(\mathcal{L})$ value, but such an implementation would only be correct in case the full $\mathcal{L}$ is

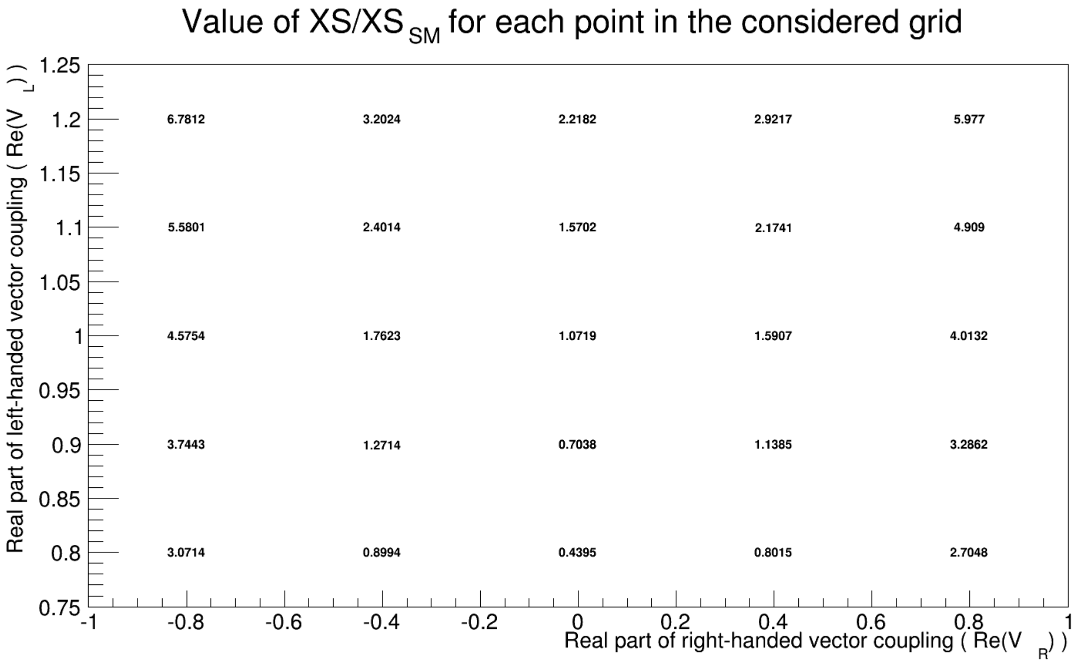


Figure 3.2: Distribution of the XS normalisation for positive semi-muonic ttbar events. As in the previous figure 3200 positive semi-muonic have been used to obtain this distribution and a single gaussian transfer function has been applied to smear the kinematics of these parton-level events.

used and not when the ln is taken. Hence the implemenation has been adapted in such a way that the logarithm is correctly taken into account[1], as given in Equation (3.4). Figure 3.2 contains the above mentioned implementation but the caption has not yet been updated.

$$\mathcal{L}_{Norm} = -ln(\sum P(y|a) * \frac{XS}{XS^{SM}}) = -ln(\mathcal{L}) - ln(\frac{XS}{XS^{SM}} * N) \tag{3.4}$$

However, even after applying this normalisation, the minimization of the $\ln(\mathcal{L})$ still does not return the Standard Model value. Therefore additional research has been performed in order to understand the origin of this discrepancy. The most likely solution is still related to the XS normalisation and is maybe expected that the current normalisation applied does not have the correct form. More detail about this matter can be found in Section 3.2.

3.2 Normalisation of Matrix Element probability

As mentioned before, the general Matrix Element formula (Equation 3.1) contains a normalisation term $\sigma(a)$. According to a mail received on 31/10/2014 this normalisation is not applied anymore automatically and should be implemented personally.

[1] *It should be checked whether this is the correct method to take into account the normalisation of the XS. Currently it has been assumed that this XS normalisation should be applied for each weight and hence is multiplied with the number of considered events. In the case that this normalisation should just be multiplied with the overall likelihood value ($\mathcal{L}$) the sum over the number of considered events drops out of the equation implying a very small influence of the XS on the* $\ln(\mathcal{L})$ *distribution.*

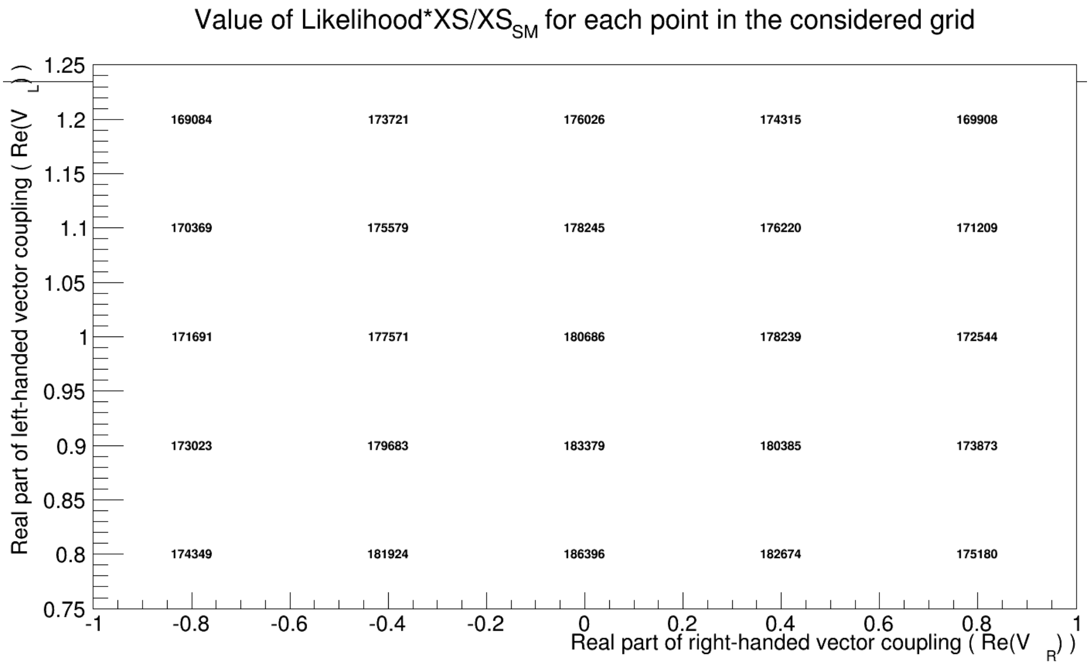


Figure 3.3: Distribution of the $\ln(\mathcal{L})$ after taking into account the XS normalisation. Again 3200 positive semi-muonic have been used and a single gaussian transfer function has been applied.

Smallness of obtained weights

All the weights calculated by MadWeight when using this AnomalousCouplings FeynRules model result in very small values and uncertainties. Up to now no weight larger than 10^{-22} have been encountered which results in a very large $\ln(\mathcal{L})$ value.

It is not clear what is the cause of these small values, but some possibilities are listed below. For the moment it is also not completely understood whether this smallness of the weights is actually related to the missing XS normalisation. First tests already have been performed in order to rule out whether such small weights are only obtained when the AnomalousCouplings FeynRules model is used and this seems not to be the case. Even when the Standard Model is used and the top mass is simulated, the calculated weights are as small as in the case of the personal FeynRules model.

- The normalisation of the MadWeight probability should still be done and is not performed within the Matrix Element Techniques formula.
- The smallness of the weight could be caused by an error inside the created FeynRules model[2].

3.2.1 Control check: Top Mass Measurement

In order to check whether this XS normalisation results in an overall influence on the obtained MadWeight output, the MadWeight generator had been used to select the most plausible top-quark mass. So instead of generator different (V_L, V_R) grid points, five

[2]Should also look for the mail where one of the MadWeight experts (Olivier/Pierre or even Celine) answered about the possible explanation for the smallness of the weight and whether this implies some wrong assumptions.

different top-quark masses have been simulated. If this would also return a completely wrong, it would be an important indication of an important issue with MadWeight. This because MadWeight has already been used succesfully to measure the top-quark mass.

In order to compare the top-quark mass result with the (V_L, V_R) one, a couple of additional comparisons have been performed as will be discussed in detail next.

STILL TO REVIEW

Comparing SM model with AnomalousCouplings model

As a first step the Feynmann diagrams belonging to the two different models should be compared. This information can be found in the *index.html* file in the following directories (and the files should be opened using firefox on mtop since this is the only m-machine with a working browser):

$/AnomalousCouplings/MadGraph5\_aMC@NLO/madgraph5/SM\_ttbarSemiMuPlus$
$/AnomalousCouplings/MadGraph5\_aMC@NLO/madgraph5/ttbarSemiMuPlus\_QED2$

Comparing SM cross section with MassiveLeptons cross section

In order to be sure that both models have the same Standard Model base, the cross sections for both models have been compared. This resulted in an unexpected outcome, namely that the obtained cross sections differ significantly depending on which MadGraph version is used to generate the considered events. A summary can be found in Table 3.2.

m_t	MadGraph aMC@NLO		MadGraph v155	
	SM model	MassiveLeptons model	SM model	MassiveLeptons model
153	9.23 pb	9.645 pb	6.692 pb	6.984 pb
163	11.12 pb	11.63 pb	7.844 pb	8.199 pb
173	12.98 pb	13.54 pb	8.897 pb	9.281 pb
183	14.77 pb	15.4 pb	9.884 pb	10.3 pb
193	16.5 pb	17.22 pb	10.78 pb	11.25 pb

Table 3.1: Cross section values for semi-muonic (+) ttbar decay obtained using two different MadGraph versions.

From this table can be seen that there is, for both considered MadGraph versions, a small difference between the SM FeynRules model and the MassiveLeptons one. This could be caused by the different treatment of the leptons. In the SM model they are considered to be massless while in the MassiveLeptons one they are defined to have their actual mass.
A larger differrence occurs when both MadGraph versions are compared. From the answer received by Olivier it is not clear whether this difference is worrysome or could be explained by the LO theoretical uncertainties. Should also be investigated whether this difference is related to the NLO behavior of the newest MadGraph version. In case the MadGraph v155 version is not up to NLO a difference in cross section is definitely expected.

Understanding why Top Mass simulation does result in correct Likelihood minimum

When using the *MadGraph5_aMC@NLO* MadWeight version is used to scan over the different top mass values the correct minimum is obtained directly, so without any normalisation or acceptance influences. This simulation only uses the Standard Model information and doesn't consider the AnomalousCouplings FeynRules model. The results can be found in Table **??**

m_t	$\ln(\mathcal{L})$	
	MW_aMC@NLO	MW_aMC@NLO ML model
153	196 499	118 238
163	188 002	114 778
173	181 027	112 803
183	186 284	116 174
193	192 926	119 717

Table 3.2: caption ..

3.2.2 Influence of the acceptance term

In order to finalize the normalisation of the weight obtained from MadWeight also the influence of the acceptance term $Acc(a)$ has to be investigated. Since no reconstructed events exist are created for the different vector coupling coefficients, the influence of the applied event selection can only be applied on generator-level. This will normally introduce a slight bias since applying the reco-level cuts on generator-level will result in more stringent cuts than actually applied on the reconstructed events due to the smaller width of the kinematic distributions. However it is the only way to get an idea of the influence of the event selection and, as long as a flat dependency is found throughout the vector couplings, an acceptable one.

The event selection influence is investigated by looking both at the change in cross section and at the difference of different kinematic distributions. This first is achieved by simulating events with different vector coupling coefficients and analyze the change in cross section. This is done for 1-dimensional scans of both the left-handed as right-handed vector coupling coefficients while the other is set to its Standard Model expectation value. The results together with the procentual reduction in cross section by the application of the event selection is given in Table 3.3 and Table 3.4 for the 1D change of V_L and V_R, respectively. All the created MadGraph files are located in the following directory:

MadGraph5_aMC@NLO/madgraph5/TopMassCheckQED2_ttbarSemiMuPlus_ML

From these Tables can be concluded that the influence of the event selection on the cross section is flat and equal throughout the entire vector coupling grid. This would be a positive result since it would imply that no additional analytical function is necessary to normalise the MadWeight output. Hence the MadWeight output should only be normalized for the Cross Section influence, but not the event selection one.

The second method of analyzing the ifnluence of the event selection on the vector couplings is by comparing the kinematical distributions before and after the event selection is applied. Special attention is awarded to ensuring no difference in shape is observed for

Top quark mass	1D change of Re(V_L)		
	All events	Reco p_T cuts applied	Reduction (%)
(0.8, 0.0)	3.62605 pb	0.9423 pb	25.99
(0.9, 0.0)	5.81248 pb	1.51 pb	25.98
(1.0, 0.0)	8.85979 pb	2.30454 pb	26.01
(1.1, 0.0)	12.96357 pb	3.37064 pb	26.00
(1.2, 0.0)	18.3674 pb	4.768 pb	25.96

Table 3.3

Top quark mass	1D change of Re(V_R)		
	All events	Recop_T cuts applied	Reduction (%)
(1.0, -1.0)	37.7415 pb	11.98 pb	31.74
(1.0, -0.5)	14.5606 pb	4.466 pb	30.67
(1.0, 0.0)	8.85979 pb	2.661 pb	30.03
(1.0, 0.5)	13.1236 pb	4.04 pb	30.78
(1.0, 1.0)	33.1415 pb	10.61 pb	32.01

Table 3.4

the different vector coupling coefficients after the event selection is applied. If this would be the case the influence of the event selection wouldn't be flat as suggested by the change in cross section. However, comparing all kinematic distributions indeed suggests that no significant change is observed when comparing the different kinematic distributions. The full list of distributions can be found in the directory listed below, but the ones for the $\cos\theta^*$ variable, the p_T distribution for the lepton, for the b-quark originating from the hadronically decaying top quark and for the down-quark originating from the W-boson decay are given in Figure 3.4, Figure 3.5, Figure 3.6 and Figure 3.7, respectively.

$.../ThesisSubjects/AnomalousCouplings/KinematicDistributions\_AcceptanceTerm\_Dec2014$

From the distributions given above can easily be concluded that some shape difference are visible for the 1D-variation of the right-handed vector coupling. However for the left-handed vector coupling V_R no influence is visible, besides some minor statistical fluctuations, when varying the value of V_R. It is also clear that the influence of the V_R 1D-variation is not the same for each of the kinematical distributions, and even negligible for the p_T distribution of the b-jet originating from the hadronically decaying top quark. Since the shift of the distribution is different for the p_T distribution of the lepton and the down-type quark of the W-boson, the result is still in agreement with the flat change in cross section. This because the additional events for one distribution are balanced out by the reduced number of events for another distribution resulting a net effect of zero and hence a flat behavior troughout the entire vector coupling coefficients grid.

3.2.2.1 Understanding 1D-variation of V_L

The influence of the variation of the right-handed vector coupling V_R on the kinematic distributions was rather satisfactory and in some way agreeing with expactations. However the same is definetely not true for the variation of the left-handed vector coupling

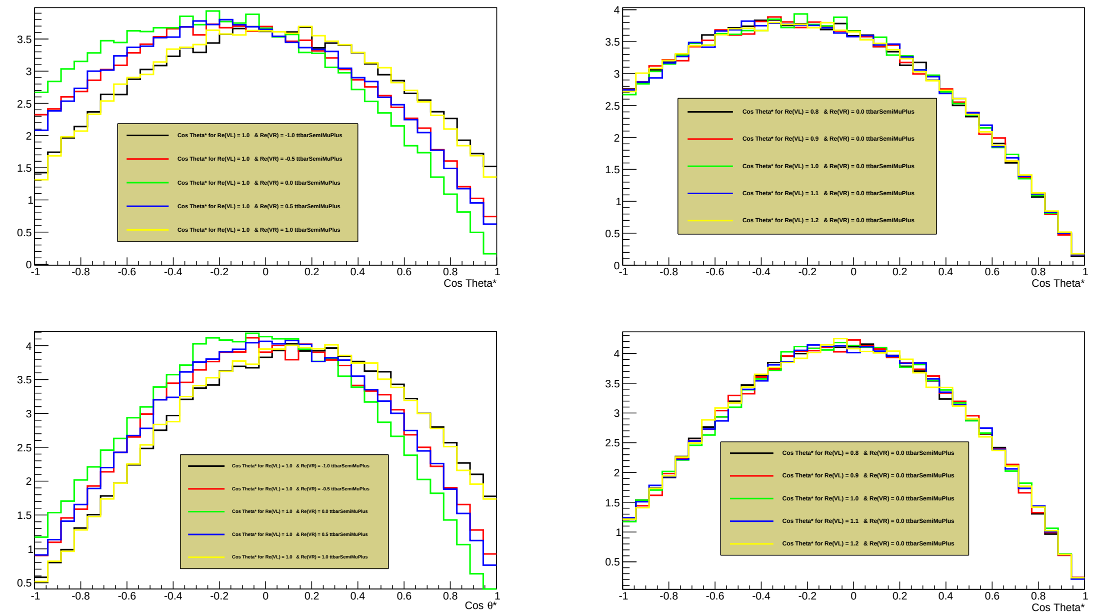


Figure 3.4: Distribution of $\cos\theta^*$ variable for both 1D-variation of the vector coupling coefficients. The top figures depict the distributions before the application of any event selection while the lower ones show the same distribution but after the reco-level p_T cuts have been applied.

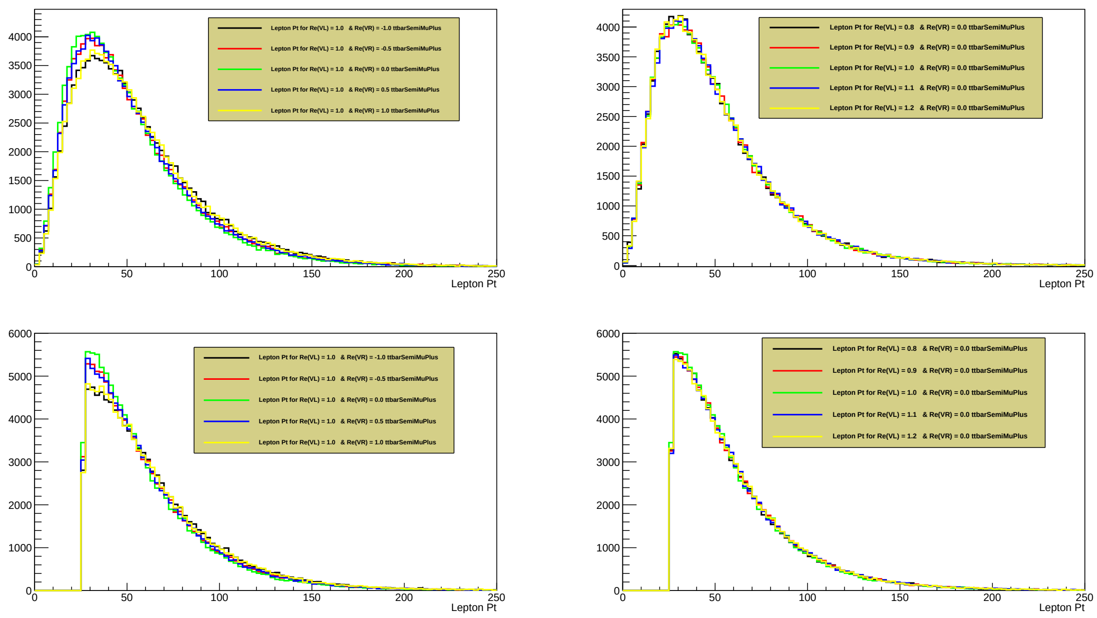


Figure 3.5: Distribution of transverse momentum of the lepton for both 1D-variation of the vector coupling coefficients. The top figures depict the distributions before the application of any event selection while the lower ones show the same distribution but after the reco-level p_T cuts have been applied.

V_L. On the contrary, it can even be concluded that the obtained result was a complete surprise and needs to be understood as soon as possible in order to keep confidence in the created FeynRules model. This because a possible explanation can always be a wrong

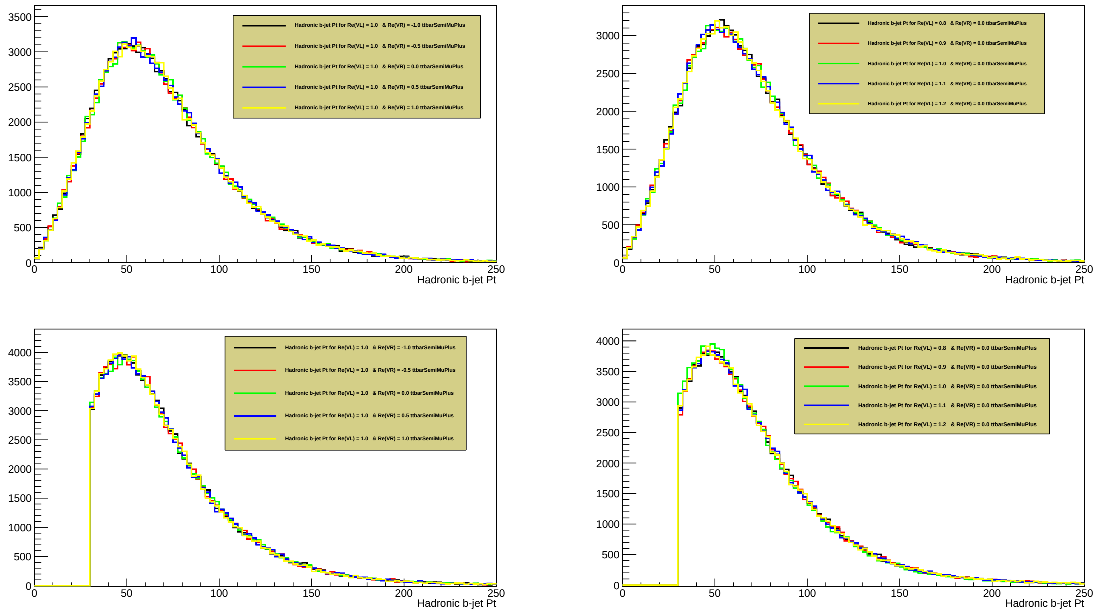


Figure 3.6: Distribution of the transverse momentum of the b-quark originating from the hadronically decaying top quark for both 1D-variation of the vector coupling coefficients. The top figures depict the distributions before the application of any event selection while the lower ones show the same distribution but after the reco-level p_T cuts have been applied.

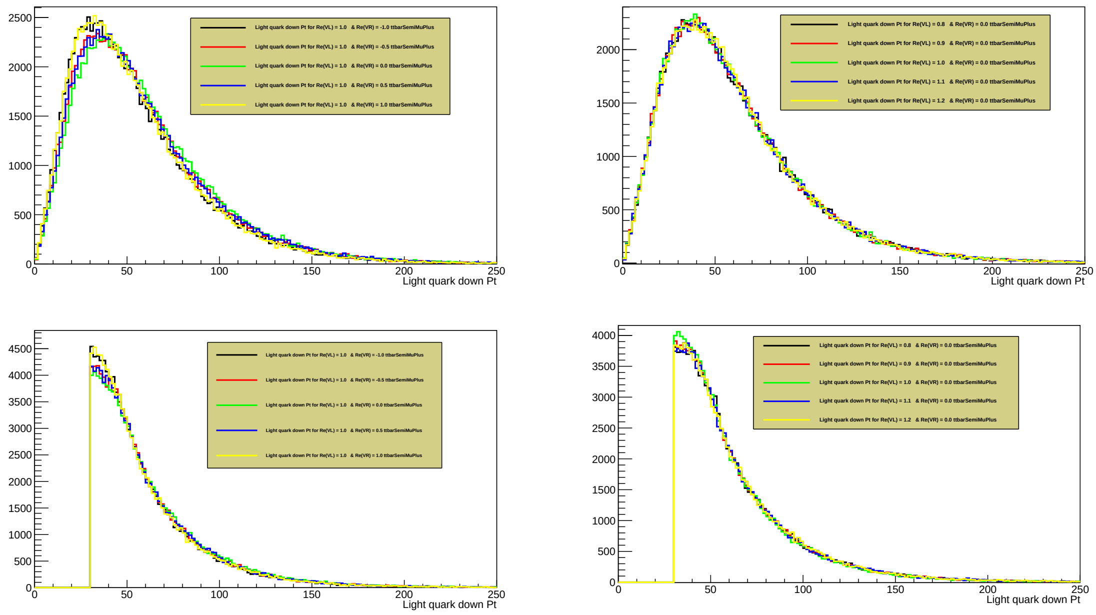


Figure 3.7: Distribution of the transverse momentum of the down-type quark originating from the W-boson for both 1D-variation of the vector coupling coefficients. The top figures depict the distributions before the application of any event selection while the lower ones show the same distribution but after the reco-level p_T cuts have been applied.

configuration of the anomalousCouplings FeynRules model with would result in a major setback for the analysis and the tight time schedule.

Therefore new MadGraph files, and kinematic distributions, have been created with the same 1D-variation of the left-handed vector coupling V_L but in stead of setting the V_R value equal to its Standard Model expectation of 0 it was set to 0.2. The reason behind this different value for V_R is the idea that when the right-handed vector coupling is excluded from the full Wtb Lagrangian any change of V_L only affects the $|V_{tb}|$ value and hence the cross section. This would imply that no real change of physical concepts is done since the Lagrangian is in a sense unchanged because no mixing of the different vector couplings appears.
The results for this 1D-variation can be found in Table 3.5 and in Figure 3.8, and shown unfortunately identical results to the 1D-variation of V_L with V_R fixed to 0.

Top quark mass	1D change of Re(V_L)		
	All events	Reco p_T cuts applied	Reduction (%)
(0.8, 0.2)	4.223 pb	1.201 pb	28.43
(0.9, 0.2)	6.615 pb	1.882 pb	28.45
(1.0, 0.2)	9.995 pb	2.787 pb	27.88
(1.1, 0.2)	14.46 pb	4.043 pb	27.96
(1.2, 0.2)	20.26 pb	5.695 pb	28.11

Table 3.5

The obtained results for the 1D-variation of V_L with V_R fixed to 0.2 seems to suggest that either the left-handed vector component of the full Wtb Lagrangian only influences the cross section and alters in no way the kinematic distributions of the decay particles or otherwise that this component is wrongly implemented in the created FeynRules model. The following test which has been performed is looking at a larger 1D-variation of V_L while still keeping the V_R component equal to 0.2. This resulted in a slightly unexpected outcome, as can be seen in Figure 3.9, since some deviation of the kinematic distribution is found for the configuration where V_L is equal to 0. However there is no difference between the four remaining configurations considered for the large V_L 1D-variation.
This is surprising since there was actually some hope that the lack of influence on the kinematic distributions when varying the V_L component could be caused by the small variation applied. This was motivated by the small influence when changing the V_R component to the boundaries of the coupling coefficients grid while originally the V_L coupling was contained within a small region due to its precise measurement.
However the result seems to suggest that there is really no influence on the kinematic distributions and that only the V_R component is responsible for these distortions. Probably a similar result would have been found when a different value of V_R would have been used.

3.2.2.2 Fixing normalisation for 1D-variation of V_R only

Since the above mentioned results seem to predict that the left-handed vector coupling actually only influences the cross section and not the kinematics, it has been decided to ensure a correct cross section normalization for the 1D-variation of V_R only. This because the influence of the left-handed vector coupling

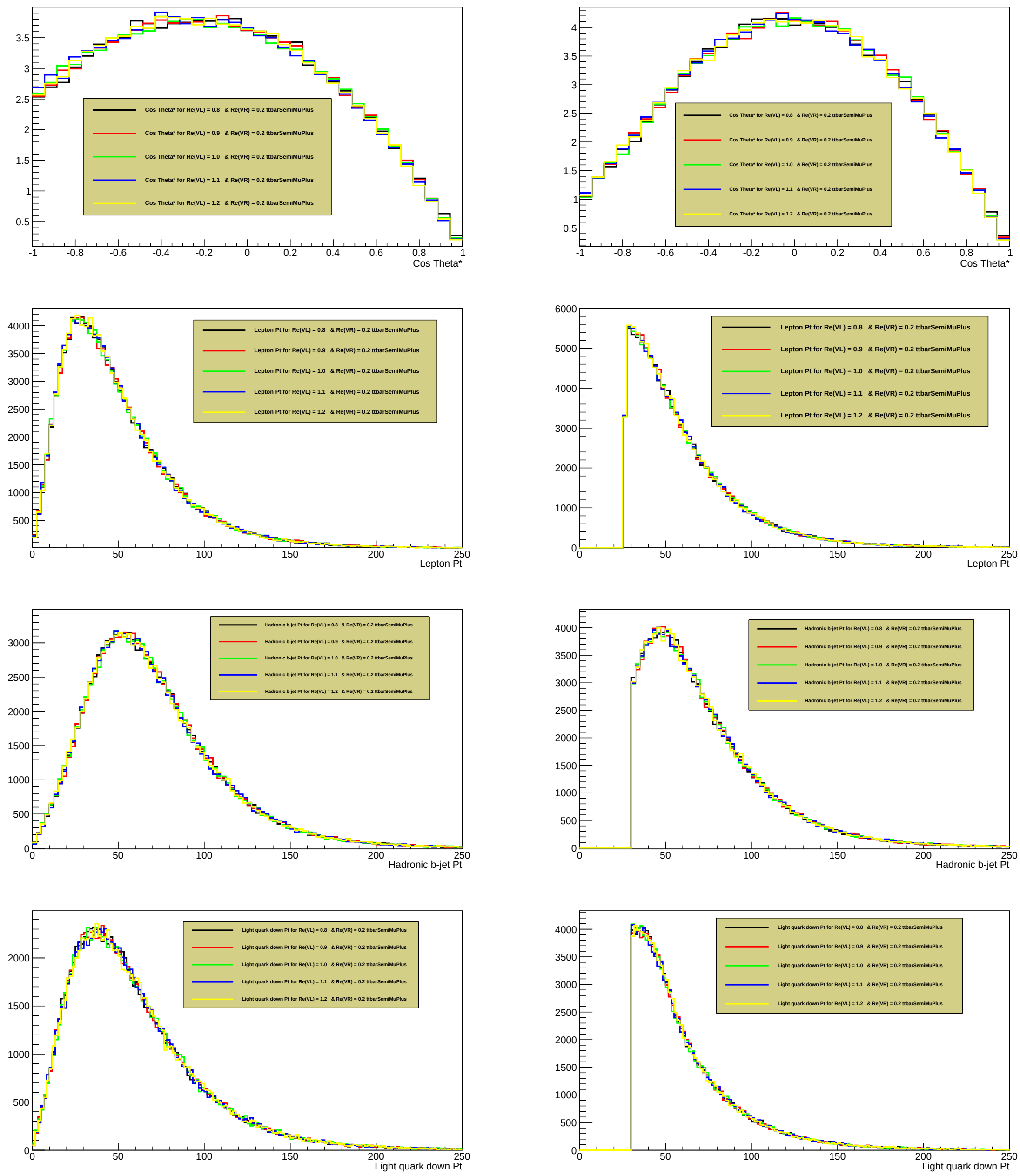


Figure 3.8

<u>Remark:</u> Correct way of thinking?

Isn't it just the opposite normalisation which should be corrected? The left-handed vector coupling only influences the cross section so the log(likelihood) should actually be normalized in such a way that it results in a flat distribution for the 1D-variation of V_L. This is which is currently still not correct because of the large difference in relative cross section for values with V_L smaller than 1.0. While I think that the 1D-variation will, just as was the case for the top quark mass, result in the correct minimization of the log(likelihood) distribution. This because both the influence of this 1D-variation on the kinematic distribution and the variation of the cross section is symmetric around the Standard Model expectation value of 0.

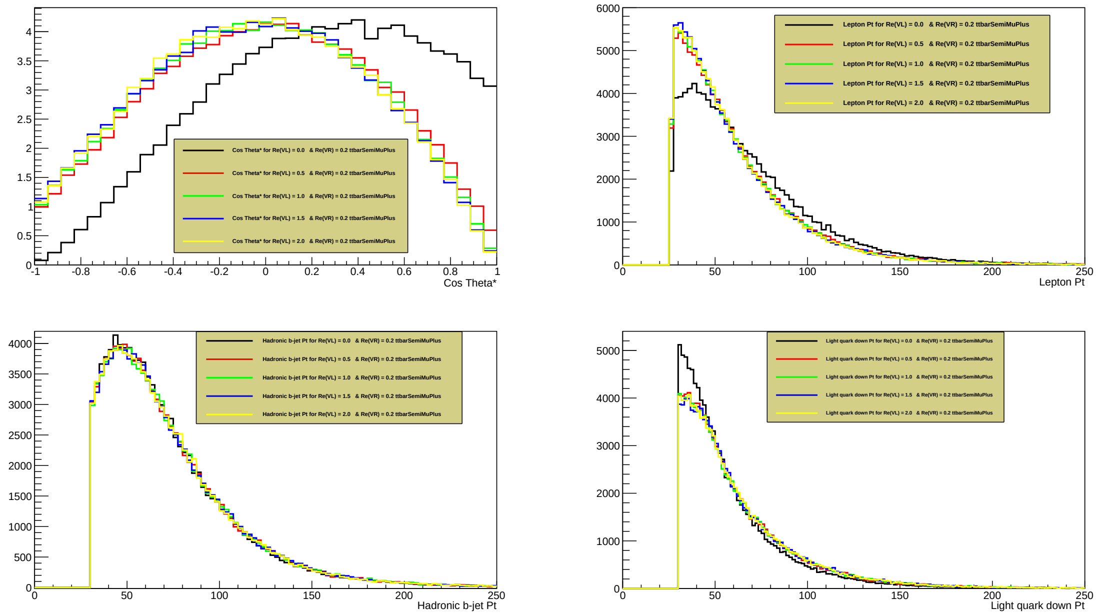


Figure 3.9

Possible suggestion could be to check the influence of another coupling constant, for instance the imaginary part of the left-handed vector coupling. The distinct difference between the left-handed vector coupling and all the other coupling parameters implemented in the FeynRules model is its Standard Model value. The left-handed vector coupling is the only parameter which is supposed to contribute to the Wtb Lagrangian, and hence the only one different from 0. Therefore it could be a possible test of the created FeynRules model to check both the variation of the cross section as the influence on the kinematic distributions when a different coupling constant is considered. If it would only be that the real part of the V_R doesn't contribute to the kinematics of the decay particles while all the other ones do have a distinct effect, it could be a clear indication that the created FeynRules model is not capable of dealing with coupling constants higher than 1.

Thinking $\cdots$

However this last conclusion is not completely correct because then a possible effect would have been visible when the V_L variable was varied in a larger range. In this case also the configurations $V_L = 0.5$ and $V_L = 0.0$ was compared to the Standard Model expectation of 1.0 but no effect was found. So there is more behind the non-influence of the left-handed vector coupling V_L than just a problem with dealing with coupling coefficients larger than 1.0 $\cdots$.

3.3 Cross Section distribution for new grid

Remark: Cross section comparison

Need to check how it is possible that the XS values in the XS grid are identical for both versions (first check whether this is indeed the case), but are different for the top quark mass simulation ... Table 3.2.

Chapter 4

Transfer Functions

In order to obtain reliable MadWeight results, the Transfer Functions which link the reconstructed energy distribution with the actual energy distributions should be taken into account. In the case of generator-level events this is less relevant since no significant smearing of the energy is expected. However in order to avoid any bias related to the used Transfer Functions it is prefered to use the constructed Transfer Functions but with reduced width.

In this analysis it has been chosen to use a double Gaussian Transfer Function. This type of function is succesful in describing both the Gaussian distribution of the peak of the distributions, but also correctly takes into account the tail. This is prefered to a single Gaussian which only fits the peak and just discards the tail of the distribution. However, as can be seen from some of the distributions further in the text, the tail can become rather significant depending on the amount of available statistics. The following equation, Equation 4.3, gives explicitely the functional form of the double Gaussian fit and the p_T-dependent "calorimeter" formula. As can be seen from this equation, the name calorimeter should not be taken to literally since it is only the actual calorimeter p_T-dependency in the case of the σ-parameters (2&5) of the two Gaussian fits.

$$a_3 \times e^{-\frac{(x-a_1)^2}{2*a_2^2}} + a_6 \times e^{-\frac{(x-a_4)^2}{2*a_5^2}} \tag{4.1}$$

$$a_{x,0} + a_{x,1} \times \sqrt{p_T} + a_{x,2} \times p_T \quad (\text{ for x} = 2 \ \& \ 5 \ ) \tag{4.2}$$

$$a_{x,0} + a_{x,1} \times p_T + a_{x,2} \times p_T^2 + a_{x,3} \times p_T^3 + a_{x,4} \times p_T^4 \quad (\text{for x} = 1,\ 3,\ 4 \ \& \ 6) \tag{4.3}$$

The method to obtain the parameters describing the energy smearing is (partly) explained in the PhD Thesis of Arnaud Pin[1].

The used calculation of the Transfer Functions is based on the code received from Petra and Lieselotte, which was used in the Master thesis of Lieselotte[2].Detailed information can be found in the following directories:

$$/home/annik/Documents/Vub/PhD/ThesisSubjects/AnomalousCouplings/$$

[1]It should be noted that in his thesis (available on *cp3.irmp.ucl.ac.be/upload/theses/phd/*), and in the current double Gaussian transfer function syntax in MadWeight, only 5 parameters are used. The narrow gaussian distribution doesn't have a normalisation parameter in front, but is normalized afterwards in the MadWeight code.

[2]Additional feedback was also received from Arnaud, but he was using a completely different unbinned likelihood method. Since this only arrived after a couple of weeks waiting, it was decided to continue with the binned likelihood method used by Petra and Lieselotte.

$PrepareGenLevelRunning\_Sep2014/TransferFunctions$
$m - machinesdirectory?$

The received code is almost identical to the so-called *ROOT* class $FitSlicesY()$ but has some differences, some of them have already been changed in order to match better with the ROOT class. One of the most important differences between the two approaches was the treatment of the underflow and overflow bin. In the received code these two bins were respectively added to the first and last bin and hence included in the fitting range. This is not the desired behavior since the size of underflow/overflow bin can be relatively large compared to the first/last bin and significantly change the value of these bins. This would then imply that the position of the first and last bin is not located at the correct position and will, especially in the case of a limited number of bins, have a significant influence on the fit result. Now these underflow and overflow bins are just discarded from the fit range and will have no influence on the final result.

Another important, but useful, difference between the two methods is the number of histograms which are saved. The received code saves for each distribution which is considered, the ProjectionY distribution together with the double Gaussian fit for this bin. This can not be changed in the ROOT class which only stores the distribution of the 6 parameters of the double Gaussian fit formula.

However, even after carefully ensuring that both methods are identical the obtained results are not. Up to now it is not clear what is the reason for the discrepancy between the two results and the only way to fund out is comparing the distributions and results for a significant amount of statistics.
One possibility is the used fit ranges and number of bins. If the last bins are low on statistics their distribution might not agree with a double Gaussian distribution and hence result in a failed fit. Therefore the distribution for each ProjectionY bin is now closely studied for all of the considered histograms. This can be found is the following section 4.2.

4.1 Creation of the Transfer Function

In this section the different steps which were performed in order to built and implement the Transfer Function (TF) will be discussed in detail. The TF's used in this analysis are assumed to be uncorrelated, as shortly discussed in the PhD thesis of Arnaud. This assumption is important since otherwise it allows to built them for each variable separately, as given in Equation 4.4. The correctness of this assumption can be checked by looking at E vs θ, E vs ϕ and θ vs ϕ histograms.

$$W(E, \theta, \phi) = W(E)W(\theta)W(\phi) \tag{4.4}$$

In this analysis it has been opted to use p_T-dependent TF in stead of the generally used E-dependent ones. This results in a large amount of adaptations of the MadWeight configuration files as will be briefly explained here. Since the TF configuration files in MadWeight allow the change of the considered kinematic variables it seemed more relevant to utilize the transverse momentum of the considered partons and jets in stead of the energy. This choice was especially motivated by the fact that the used *.lhco* file was constructed to contain the transverse momentum as input variable. As a consequence the TF configuration file **file** should be adapted to use the *pt(p)* and *pt(pexp)* variables in stead of *p(0)* and *pexp(0)*.

Since MadWeight contains some hard-coded information about the kinematical information used in the calculation of the TF's, the use of p_T-dependent ones resulted in some difficulties. One important observation was the fact that as soon as the configuration files are adapted to be compatible with the p_T-dependent ones, there is no easy turning back to the original E-dependent TF's. This is not the desired behavior since it excludes the possibility of quickly testing any discrepancy between the created TF and the original ones. Hence it has been decided to include the relevant MadWeight directory on gitHub and develop two separate branches, one for each type of kinematic variable. This allows to easily switch between the TF's and conduct as much tests as desired. The corresponding branches on GitHub are **TF_EDependent** and **TF_PtDependent**.

Within these branch also all the different Event directories containing all executed tests are stored. This is used as a back-up since these results are only stored on localgrid where no back-up scheme is foreseen. The only disadvantage of this is that a clear overview of the performed changes to the configuration files is missing. **Give a short overview of what had to be changed!**

4.1.1 Used analysis files

The Transfer Functions are calculated using the simulated $t\bar{t}$ sample which will be used throughout the entire analysis. A very small nTuple is created from this simulated sample containing only the TLorentzVector information of both generated and reconstructed particles. From this the necessary diagrams, such as the 2D distributions of the p_T-, θ- and η-difference between the generated and reconstructed particle with respect to the generated value, are created and saved in the following ROOT file (located on the m-machines **or also copied locally??**):

$AnomalousCouplings/TFInformation/PlotsForTransferFunctions\_FromTree.root$

The main analyzer, called $TFFit.cc$, performs the double Gaussian fit of these 2D histograms and afterwards the E-dependent calorimeter fit. The technicalities and specific details of these fit procedures are documented in the $TFCreation$ class which can be found in the $PersonalClasses$ directory.

The results of the two consecutive fits performed on the Y-projections of these 2D distributions are stored in a different ROOT file, together with the original 2D histograms. Also the function form of the double Gaussian fit formula using the obtained fit parameters, added in order to test the robustness of the fit results outside the fitted range, can be found in this ROOT file.

$AnomalousCouplings/TFInformation/CreatedTFFromDistributions\_FromTree.root$

This analyzer also has the flexibility to perform the fit on the entire range or on pre-defined ranges set by the user. For this a separate function, called *SetFitRange*, is created where for each histogram the fit range for each separate bin can be defined. This is extremely useful to optimize the doubleGaussian fit which has to cover both the peak and the tail of the distributions in order to correctly calculate the 6 fit parameters.
Another useful aspect of this analyzer is the automatic creation of the necessary *.dat*

Transfer Function files needed for implementation in MadWeight. This is done in the *WriteTF* class and the created files are[3]:

$$AnomalousCouplings/TFInformation/TF\_user.dat$$
$$AnomalousCouplings/TFInformation/transfer\_card\_user.dat$$

This first file contains the functional form of both the E-dependent calorimeter fit and the functional form of how these 6 parameters should be included in the double Gaussian formula. Also the width for the different kinematic variables is defined within this file. For the moment the method used in the Transfer Functions already implemented in MadWeight is followed, implying that the width of the Transfer Function is defined as the maximum of the σ-parameter of the two Gaussians considered in the double Gaussian fit. The only difference is that in stead of the generated kinematic information, which is the variable on the abscissa of the considered 2D histograms, the reconstructed one is used.

The second file contains the actual values of the different fit parameters for all the considered kinematic variables and particle types. Therefore this file is an extensive list of values to which is referred in the previous $TF\_user.dat$ file. For each particle type and kinematic variable the numbering used should be unique such that the correct values are implemented in the functional forms of the used fit formulas.

Separating narrow and wide gaussian

In order to differentiate between the narrow and wide gaussian both the amplitude and the σ-parameter should be compared. However the most important parameter is the latter one, σ, since this represents the width of the corresponding Gaussian distribution. Nevertheless it is expected that the distribution with the narrowest distribution also has the highest peak.
In order to easily compare the two distributions a stacked canvas is added to the ROOT file which shows both distributions for different $p_{T,gen}$ values together. The distinction between narrow and wide gaussian distribution is currently being made by the size of the σ variable. Hence the histogram with the narrowest distribution is plotted in red while the widest one is plotted in green. Such a canvas is made for each of the considered 2D-histograms. They should be analyzed in detail in order to understand the correctness of the double Gaussian fit applied for the creation of the Transfer Functions. For the moment there a still a couple of 2D-histograms which don't show the expected behavior. **Additional investigation of these stacked canvasses should be performed as soon as possible.**

Remark: Should check whether this splitting in narrow and wide gaussian is actually necessary for the MadWeight implementation. MadWeight only seems to need the general fit formula and doesn't need to know which of the two distributions is the narrow and which is the wide one.

Importance of start values

Since the double Gaussian fit really needs accurate information of both the peak and the tails, detailed review of the start values for each of the 6 fit parameters significantly improves the success rate of the fitting method. Hence special care should be awarded

[3] Currently two different files are created, one for separate η bins and one for all of the events.

to ensure the correctness of these start values by comparing the fit distributions for the different particles.After quite a while it is possible to quickly see whether the fit distribution has the expected shape and whether the used start values can be considered as stable. The start values used for the different 2D histograms is given in Table 4.1.

2D histogram	Used start value for fit parameter					
	First (narrow) gaussian			Second (wide) gaussian		
	Mean a_1	Sigma a_2	Amplitude a_3	Mean a_4	Sigma a_5	Amplitude a_6
b-jet $\Delta\phi$	0.0002	0.022	8000	0.0002	0.06	3000
b-jet Δp_T	10	-12	20000	13	10	-5000
b-jet $\Delta\theta$	0	0.013	6000	0	0.04	2000
light jet $\Delta\phi$	0	0.022	8000	0.0004	0.002	3000
light jet Δp_T	0	8	4000	0	12	4000
light jet $\Delta\theta$	0	-0.014	6000	0	-0.05	2000
electron $\Delta\phi$	0	0.0012	1500	0	0.006	600
electron Δp_T	0	0.9	1500	0	-2	600
electron $\Delta\theta$	0	0.0013	2500	0	0.007	600
muon $\Delta\phi$	0	0.0004	800	0	0.0026	600
muon $\frac{1}{\Delta p_T}$	0	0.0003	2000	0	0.0006	500
muon $\Delta\theta$	0	0.002	500	0	0.0004	500

Table 4.1: caption ... **Need to make sure that the narrow and wide gaussian is always the same ... Otherwise are the start values not correct for the different eta-bins ...!!**

4.1.2 Applied $|\eta|$ binning

Since the kinematic variables tend do depend on the pseudorapidity η the considered 2D histograms are created for four distinctive $|\eta|$ regions. It has been chosen to split the barrel region into three separate bins while the entire endcap region is contained within one single bin. The chosen binning is given in Table 4.2 together with the percentage of events present in each of the considered $|\eta|$ bins. This clearly shows the lower statistics available in the endcap region which results in larger difficulties of properly reconstructing the fit parameters in this region. This is shortly discussed below.

	$\|\eta\| \leq 0.375$	$0.375 < \|\eta\| \leq 0.75$	$0.75 < \|\eta\| \leq 1.45$	$1.45 < \|\eta\| \leq 2.5$
Relative # events	26.21 %	23.91 %	32.54 %	17.34 %

Table 4.2: Different $|\eta|$ bins used for the Transfer Function creation. It is important to note that the η-values used for this binning are the reconstructed ones since generated values could be higher than the 2.5 cut-value.

The analyzer mentioned above is developed in such a way that both the fit results for all events as the results for the four separate $|\eta|$ bins are stored together. Therefore all the results can always be compared in the created ROOT files and in the distinct *.dat* files.

One important difference between the 2D-distributions containing all events and the 2D-distributions specific for one of the $|\eta|$ bins is the number of bins used. Since the statistics is significantly lower for the $|\eta|$ specific histograms, the predefined bin number is lowered with 25%. This ensures a more stable tail for the distribution and still a correct reconstruction of the peak. However for the last $|\eta|$ bin considered, the endcap part, a slightly different method is used. Since the statistics is much lower in this part of the detector the used range had to be slightly stretched in order to ensure a nice overview of the tails. This is necessary for the correctness of the double Gaussian fit. Therefore the range of the abscissa is enlarged on both sides with 20% and the used number of bins for this axis is identical to the one used for the overall 2D-distribution.

Coping with low statistics in last $|\eta|$ bin
Solution : Excluding bins! + combining bins
Explanations!!

4.1.3 Implemenation in MadWeight – NEED TO UPDATE!

All the possible Transfer Functions which are implemented in MadWeight can be found in the $Source/MadWeight/transfer\_function/data$ directory. Any file can be added to this list and used within MadWeight.
The relevant files used for the creation of the Transfer Functions are given below. The first one is the translation of the used $TF\_user.dat$ into a MadWeight readable file which can be implemented. The second file is only relevant around line 292 where the function used for the Transfer Function creation is explained. This is the general MadWeight constructor file where all the different MadWeight functions are defined.

$$Source/MadWeight/transfer\_function/transfer\_function.f$$
$$bin/internal/madweight\_interface.py$$

What about call_tf.f file which has to be changed every time a new TF is initialized!
The implementation in MadWeight should be done in such a way that the value of the outermost bins is used for all the p_T values outside the fitted region. This is necessary since the extrapolation using the obtained fit parameters doesn't result in the desired double Gaussian behavior for these p_T values. For values outside the fitted region, an inverted double Gaussian distribution or a distribution with two distinct peaks occurs rather often. This can be seen in Figure 4.1

4.2 Obtained distributions

Each of the considered histograms is a 2D histogram where the abscissa represents the transverse momentum of the generator level parton and the ordinate the difference between the generator level parton and the reconstructed matched particle. This is done for the difference in transverse momentum and in θ and ϕ angles.
All the interesting histograms can be created automatically, for each of the desired $|\eta|$ bins separately, using the following ROOT analyzer:

$$AnomalousCouplings/TFInformation/FitDistributions/SaveFitHistograms.C$$

This analyzer is able to create each of the histograms separately, both as *.pdf* and *.png*[4], and automatically saves all the histograms of one type in a large stacked canvas.

[4] *.png* files are less interesting since with the package *graphicx* *.pdf* figures can be included in $LaTeX$.

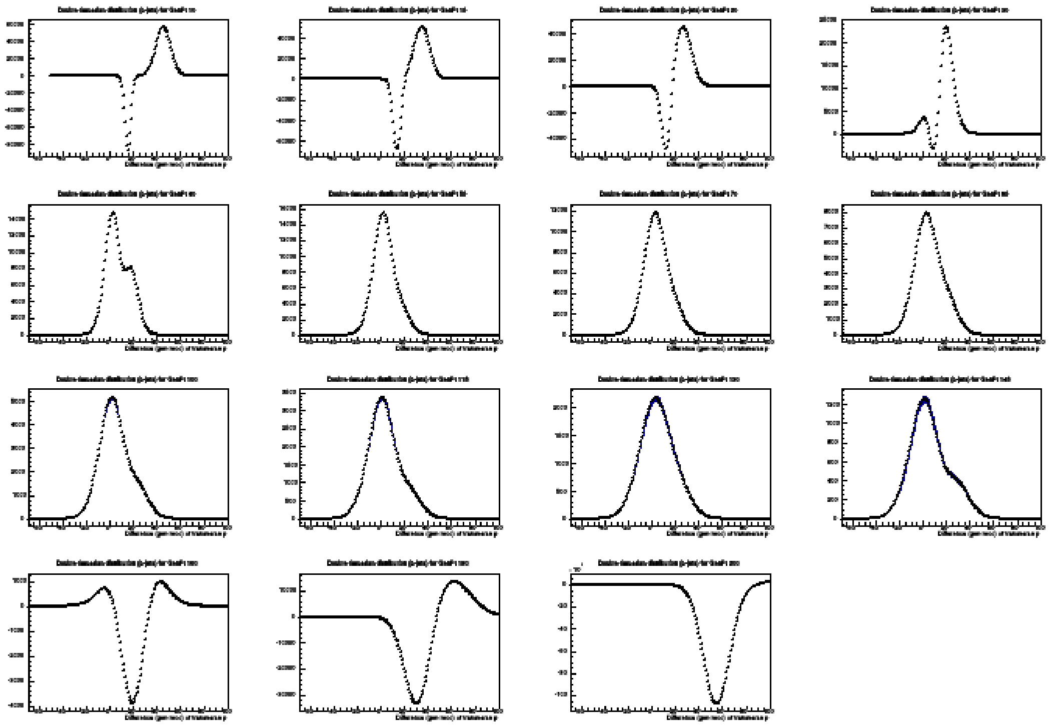


Figure 4.1: Extrapolation obtained using the fit parameters of the double Gaussian functional form. Values outside the fitted range show a distinctly different shape with respect to the ones actually fitted. *Maybe consider to lower the number of bins and make the titles more visible (Only p_T cut value is really important!)*

This allows to quickly see the used 2D distributions for each of the particle types (b-jets, light jets, electrons and muons) and the three kinematic variables (p_T, η and ϕ). This stacked canvas is shown in Figure 4.2. Also the general behavior of each of the fitted diagrams together with the overal χ^2 distribution is created for each 2D histogram as can be seen from Figure 4.3, showing this for the p_T distribution of the b-jets. Finally the distribution of the 6 double Gaussian fit parameters together with the fit result of the E-dependent calorimeter formula is also collected in a stacked canvas, as can be seen in Figure 4.4. The 2D distributions for the difference in transverse momentum tend to show a slightly asymmetric behavior, as can be seen from Figure 4.2. This can be explained by the influence of the event selection, which has a difference effect on the generated particle than the reconstructed particle. This because a particle surviving the p_T cut actually has a different p_T value on generated level due to **bad resolution, detector effects (???)**. This effect is almost negligible for the ϕ and θ angles **(Definitely sure that this is the case ??)**.

4.3 Control checks for Transfer Functions

Compare results with normal (single) Gaussian

Still to do ...

Compare result with previous analyses

This will imply to put $p_{T,reco}$ information on the abscissa in stead of the current $p_{T,gen}$

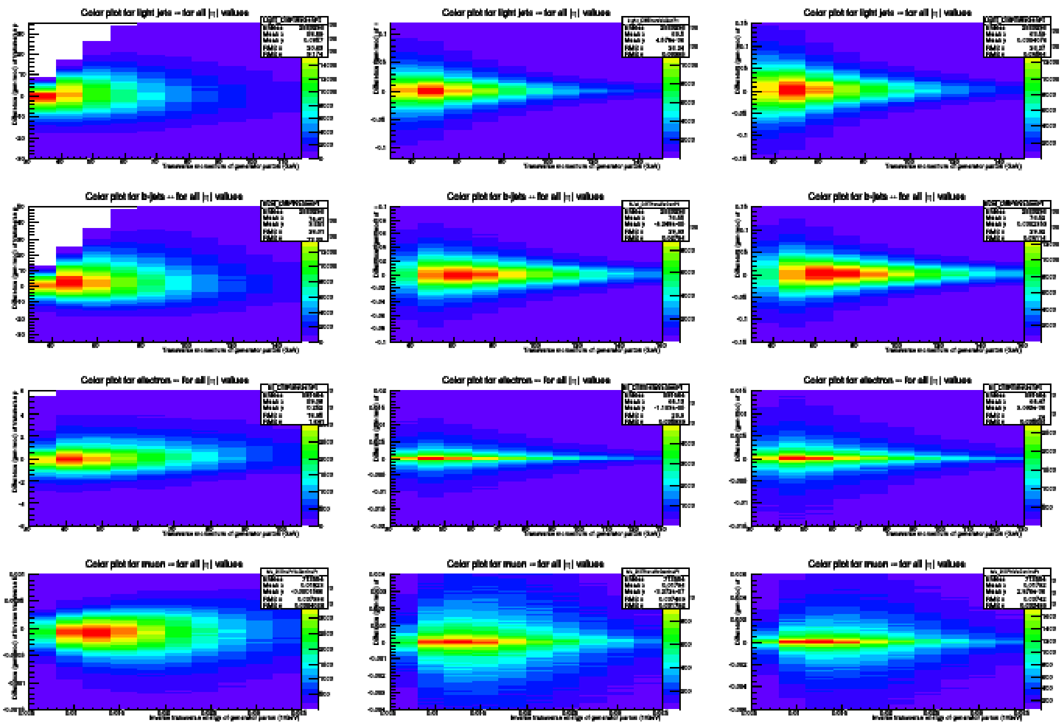


Figure 4.2: Used 2D distributions for double Gaussian fit. **IMPROVE CAPTION!**

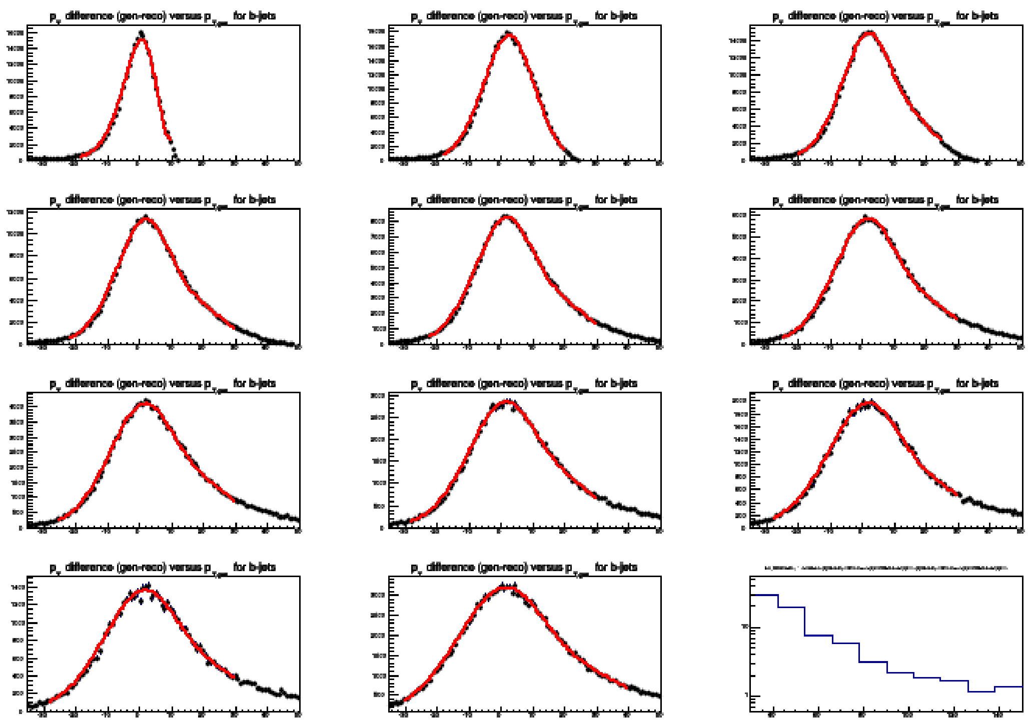


Figure 4.3: Distribution of the energy difference between the generator level parton and the corresponding light quark jet for each of the 10 considered bins and the overflow bin. All distributions were fitted with a double Gaussian function.

one. Also it will result in a huge change of the start values which are however necessary to perform a succesful double Gaussian fit.

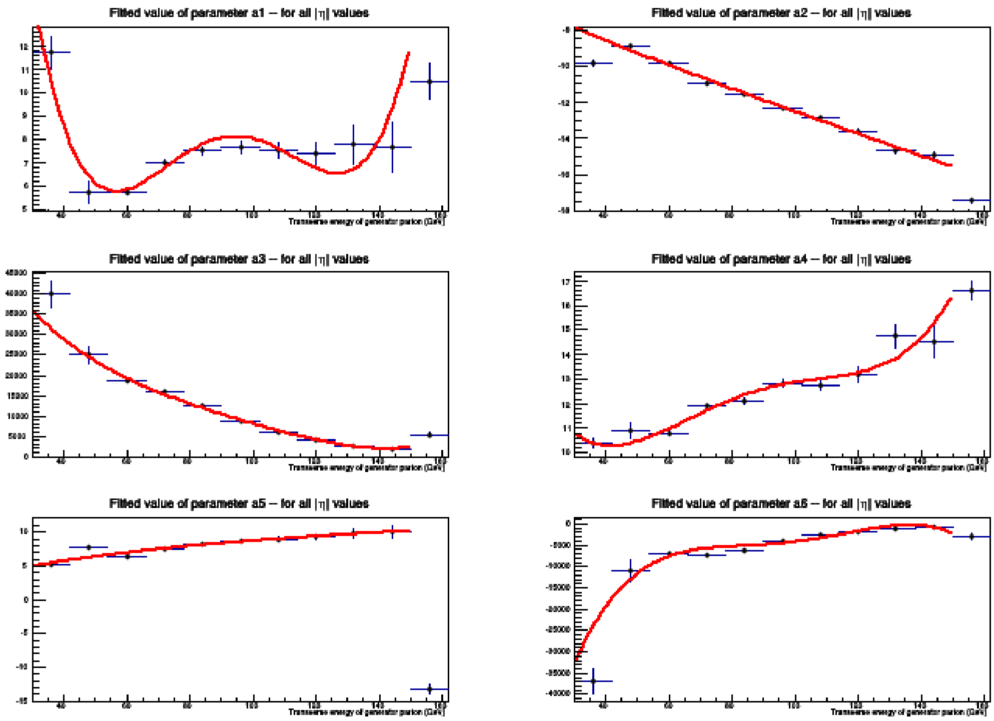


Figure 4.4: Energy dependency of the 6 parameters of the double Gaussian fit function. The result of the double Gaussian fit for each p_T bin is combined in a p_T dependent histogram and then fitted with the Calorimeter energy function as explained in the PhD Thesis of Arnaud Pin.

Compare results with predefined ROOT class

Still to do ...

Chapter 5

Preliminary Results

5.1 Results on reconstructed level

5.2 Background influence

Chapter 6

Understanding MadWeight Results

6.1 Comparing the two used MadWeight versions

Since it was found that the latest MadWeight version (aMC@NLO) resulted in many events with weight equal to 0 it was decided to compare this version with the previous one (mc_perm). It is expected that both versions result in similar weights when identical events are considered, otherwise the used version of MadWeight would have a too large influence on the analysis result.
Therefore two events which could succesfully run in the newest MadWeight version were also calculated using the old version. The obtained weights and their uncertainty can be compared in the following table and all the relevant information can be found in:

$$/home/annik/Documents/Vub/PhD/ThesisSubjects/AnomalousCouplings$$
$$/CompareMWVersions\_May2014$$

Event Number	V_L value	aMC@NLO version		mc_perm version	
		Weight	Uncertainty	Weight	Uncertainty
1	1.5	9.7610^{-28}	4.0710^{-30}	1.4410^{-26}	4.1910^{-29}
	1.0	1.9210^{-28}	8.0410^{-31}	2.8410^{-27}	8.2810^{-30}
	0.5	1.2010^{-29}	5.0310^{-32}	1.7810^{-28}	5.1710^{-31}
2	1.5	1.8610^{-23}	1.1510^{-25}	1.7710^{-24}	1.2310^{-26}
	1.0	3.6510^{-24}	2.2510^{-26}	3.5710^{-25}	2.7510^{-27}
	0.5	2.2710^{-25}	1.4710^{-27}	2.2310^{-26}	1.7210^{-28}

Table 6.1: Weight obtained from MadWeight for two specific ttbar semi-muonic (+) events. For these events the V_R was fixed to its Standard Model expectation value, which is 0, while the V_L value was varied.

Comparing these values clearly shows that there is a significant difference between the two MadWeight versions which were considered in this analysis. With some effort a general difference of a facto 10 can be identified between the two versions, with a higher weight value for the older mc_perm MadWeight version.
However when showing the relative differences between the different weights it can be seen that the behavior of these two MadWeight versions is actually very similar. Therefore the histograms below give firstly the actual weight value and secondly the weight value normalised to the weight corresponding to the coupling parameter $V_L = 0$.

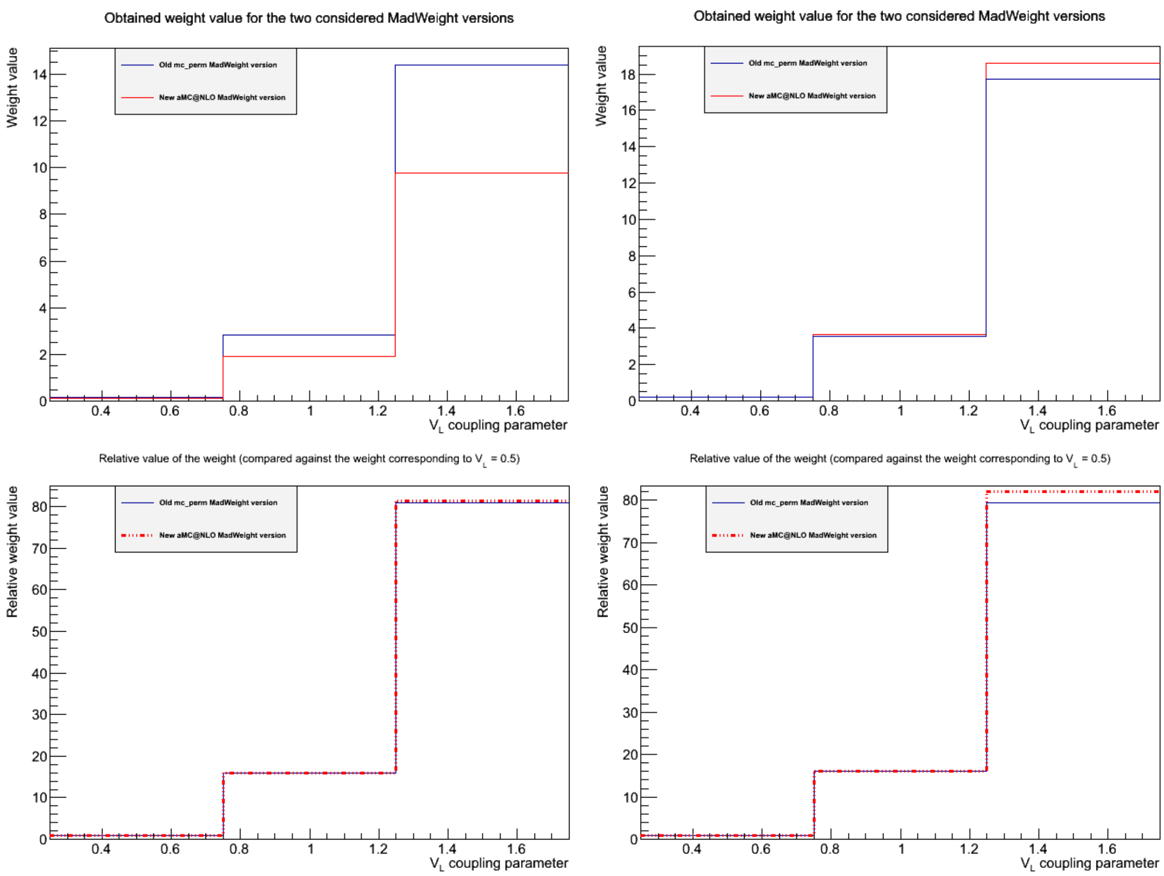


Figure 6.1: Distribution of the weights obtained from MadWeight for the two considered MadWeight versions (aMC@NLO and mc_perm) for two specific ttbar semi-muonic (+) events.

The above histograms clearly indicate that however the actual values of the weight significantly differ, the normalized results are almost identical.
This implies that the results obtained with MadWeight should always be normalized with respect to another MadWeight result (obtained using the same MadWeight version of course).

6.2 Individual weight distribution for condsidered grid-points

The first Likelihood distribution for the considered gridpoints gave rise to an unexpected result. In order to understand whether this strange behavior can be explained by a couple of events with a bad weight, the indiviudual weight distribution for a couple random events was studied. If these distributions indeed result in the correct behavior the events which influence the overal Likelihood distribution can be studied individually.
All the relevant information can be found in the following directory:

$$/home/annik/Documents/Vub/PhD/ThesisSubjects/AnomalousCouplings/ UnderstandLikelihoodDistr\_July2014$$

And the creation of the MadWeight weights together with the python scripts making the corresponding histograms can be found in:

$$/localgrid/aolbrech/madweight/ttbarSemiMuPlus\_QED2/Events$$

The relevant python scripts are the following:

- **CalculateLikelihood.py** which simultaneously creates the histograms for the XS distribution, the raw and normalized Likelihood distribution, and if required the individual weight distributions for the selected events. All these histograms are saved in the Histos.root file.

- **RemoveZeroWeightEvents.py** which removes the events with weight equal to zero from the list and saves the non-zero events in a new .out file. On the other hand the events which failed the MadWeight computation are saved on a new .lhco file together with one succesful control event and are send again through MadWeight for a new weight calculation. This should be updated since the new computation of MadWeight also results in weights equal to zero for these failing events. So should be investigated what is different about these events ...

The first three histograms show the obtained Likelihood distribution for the different gridpoints which were considered. From these can be concluded that the behavior of the Likelihood normalized with the corresponding cross section divided by the Standard Model cross section is dominated by the distribution of the normalized cross section values. This means that any small deviations of the raw Likelihood values gets washed out by the multiplication with the normalized cross section values.

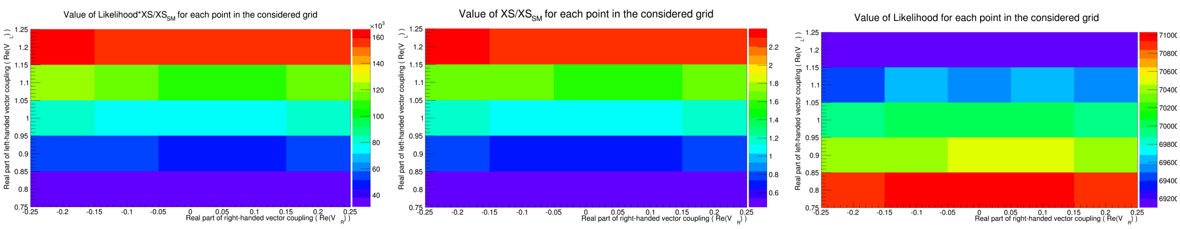


Figure 6.2: ...

The individual weight distribution for some random events can be found here.

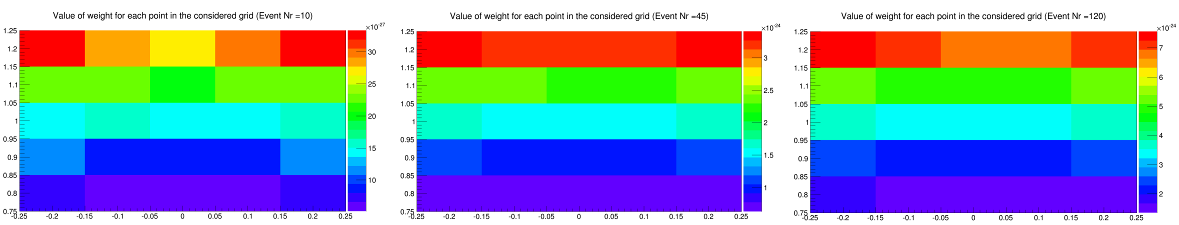


Figure 6.3: ...**Script should be ran again, and names for the axis should be added for these individual weight histograms!**

Chapter 7

Analyzing FeynRules model

7.1 Normalized coupling parameters

In order to investigate the actual influence of the value of the coupling parameters on the kinematics of the event, the considered coupling parameters should be normalized to unitarity before any hard conclusions can be made. Therefore the configurations which should be investigated in large detail are the ones for which the width of the decay remains unchanged. This is explained in detail in one of the previous sections (**??** on page **??**).

7.2 Understanding parameters larger than 1

Before starting to look at the ttbar Monte Carlo and reweighting the events with MadWeight, the created model in FeynRules should be completely understood. Special care goes out to the behavior of the kinematic distributions for values of the coupling parameters larger than 1. Since the Standard Model expectation puts the real part of the left-handed vector coupling V_L almost equal to 1, simulation should be available around this Standard Model expectation value. Therefore the created model should be able to cope with coupling parameters larger than 1.

For this reason .lhco files were generated with MadGraph with the following configuration, as mentioned in 2.4.2:

$$\begin{aligned} Re(V_L) &\in [0.7,\ 1.3] \\ Re(V_R) &\in [-0.3,\ 0.3] \end{aligned}$$

For these generated events the main kinematic distributions have been investigated. No clear difference between the behavior below and above 1 has been found.

7.2.1 Performed checks

7.2.1.1 Cross section change

RVR / RVL	-0.30	-0.20	-0.10	0.00	0.10	0.20	0.30
0.70	0.3775	0.3120	0.2724	0.3275	0.2632	0.2910	0.3436
0.80	0.5964	0.5097	0.4595	0.4385	0.4444	0.4816	0.5471
0.90	0.9011	0.7950	0.7308	0.7026	0.7103	0.3229	0.8356
1.00	1.3187	1.1874	1.1085	1.0711	1.0823	1.1317	1.2263
1.10	1.8700	1.7116	1.6154	1.5669	1.5763	1.6335	1.7522
1.20	2.5858	2.3996	2.2789	2.2200	2.2263	2.2983	2.4322
1.30	3.4896	3.2711	3.1278	3.0626	3.0655	3.1506	3.2983

Table 7.1: Cross sections for the different RVR-RVL couplings normalized to the SemiElMinus Standard Model Cross section (8.261 pb)

From this table can be seen that the cross section increases when the real component of V_L gets larger. The value of the right-handed vector coupling has only a minor influence on the cross section.

7.2.1.2 Relative increase visible in XS, but not in kinematic distributions

Since the observed model cannot represent physics at values larger than 1, one option is to look at specific fixed values of the real part of the left-handed and right-handed vector couplings. A proportional change in both of these coupling parameters should change the cross section values, but the kinematic should remain unchanged. Therefore the following configurations will be investigated:

$$\begin{aligned} Re(V_L) = 0.5 \quad &\& \quad Re(V_R) = 0.5 \rightarrow \; 2.07115\ pb \\ Re(V_L) = 1.0 \quad &\& \quad Re(V_R) = 1.0 \rightarrow \; 33.1479\ pb \\ Re(V_L) = 2.0 \quad &\& \quad Re(V_R) = 2.0 \rightarrow \; 530.027\ pb \end{aligned}$$

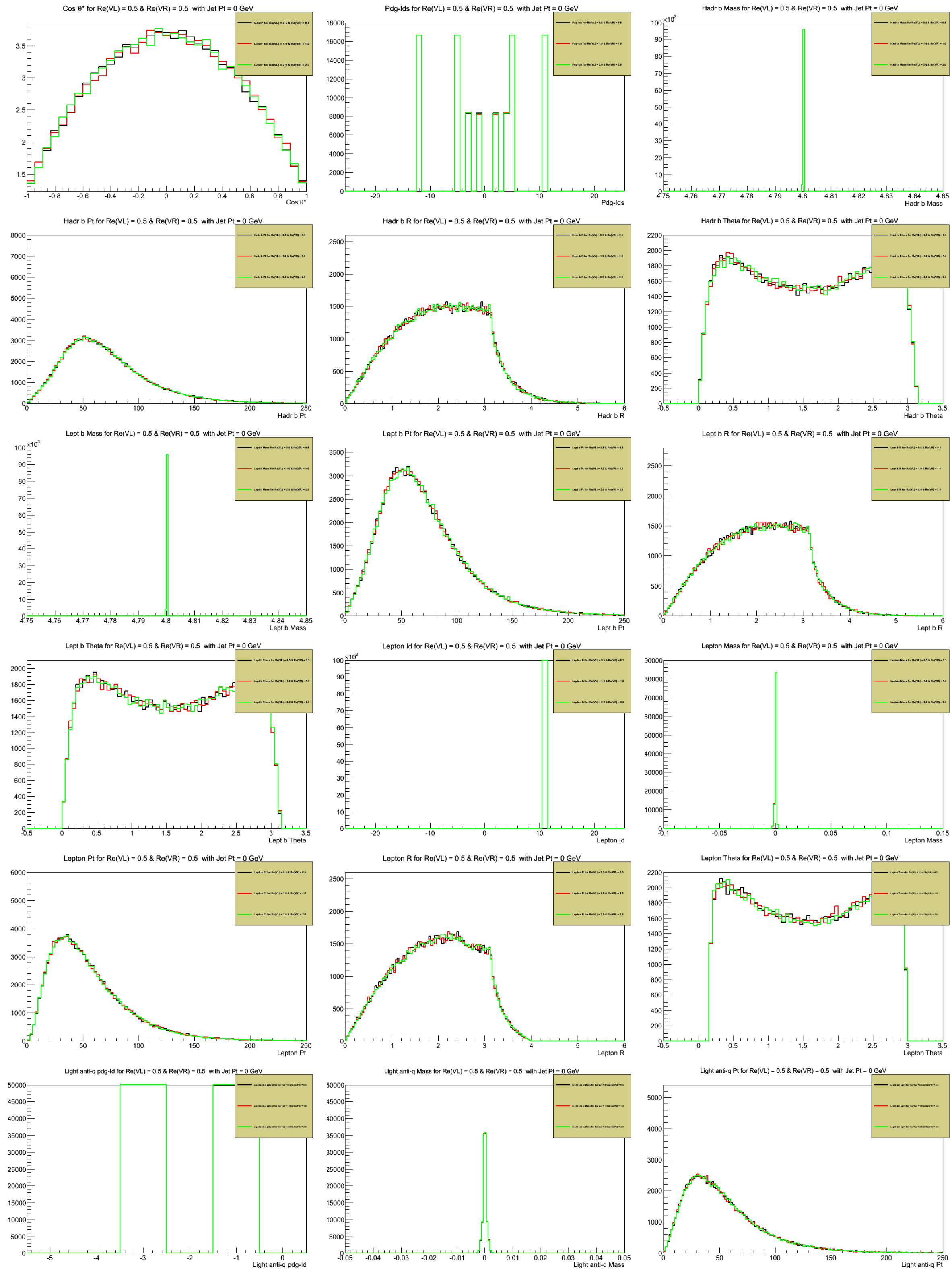
From the above numbers is clear that the Cross section becomes very large when the two coupling parameters increase. This can be understood quite easily since the second option allows much more decay options since the top quarks can decay both through the left-handed and the right-handed vector coupling side of the interaction vertex. The width of this configuration is not equal to the width of the Standard Model expectation and hence does not correspond to an actual physical solution. It is merely seen as a test of the model since the kinematics of the interaction should not differ.

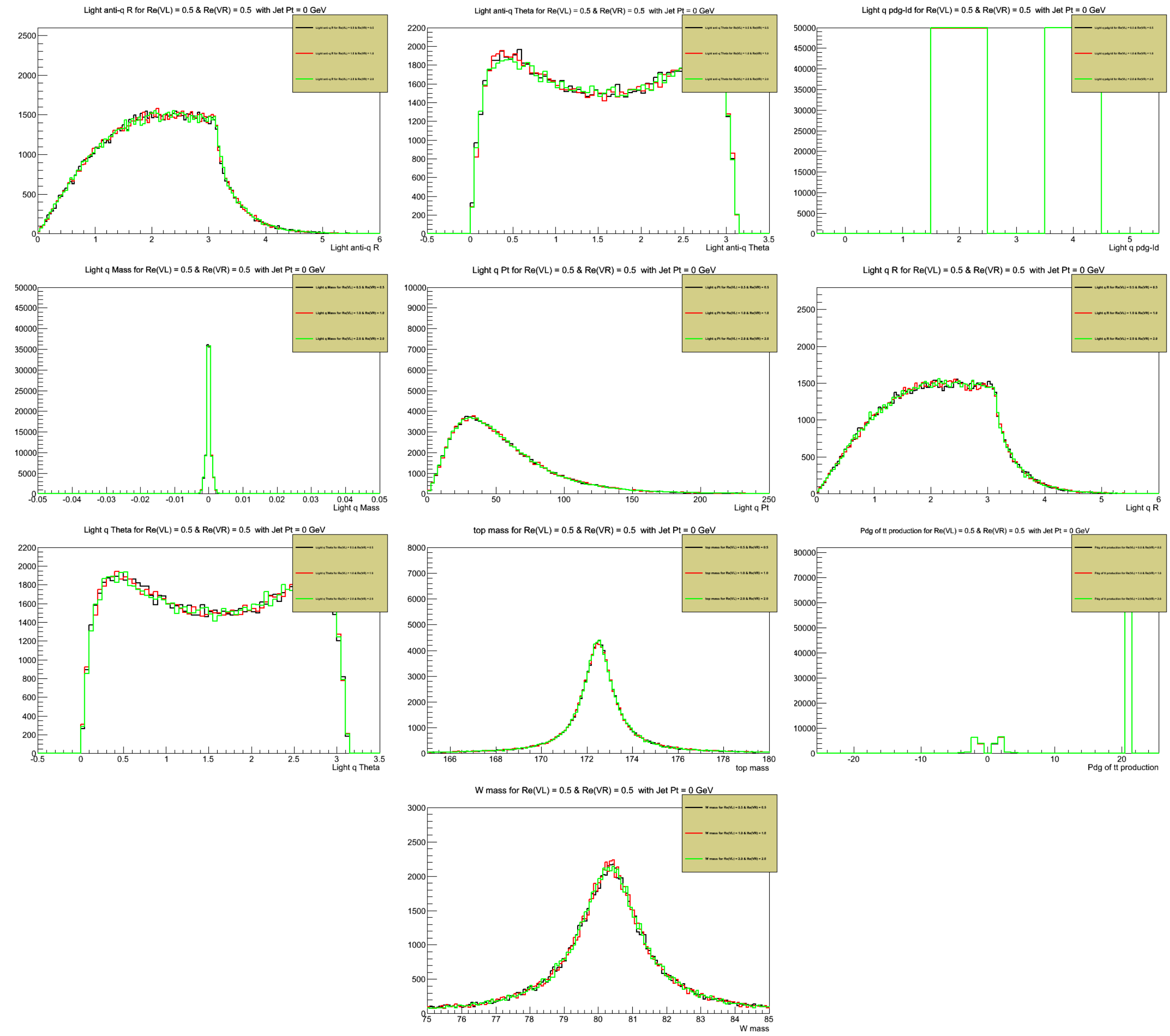
Looking at these plots clearly indicates that the kinematics doesn't change at all.
Hence the created FeynRules model is able to deal in a correct way with these coupling parameters larger than 1.

These MadGraph files have been created and can be found in:

$$/user/aolbrech/AnomalousCouplings/MadGraph\_v155/MassiveLeptons/ \\ MadGraph5\_v1\_5\_5/Wtb\_ttbarSemiElMinus/RelativeChange$$

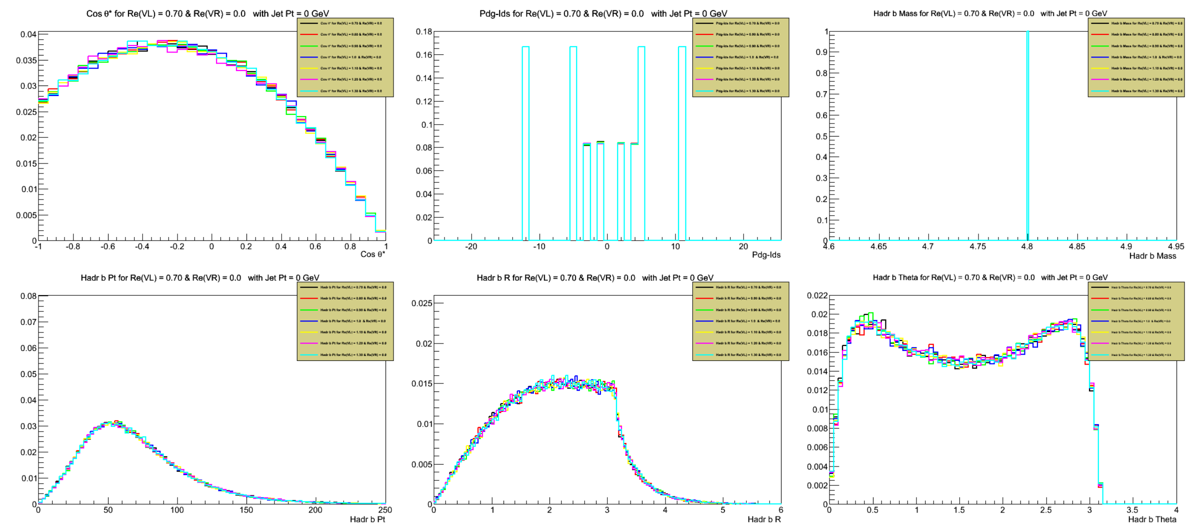
The distributions shown in this subsection are for fixed Jet Pt Cut value, set to 0. Also no Pt cut on the lepton was applied. Both coupling parameters have been changed proportionally.

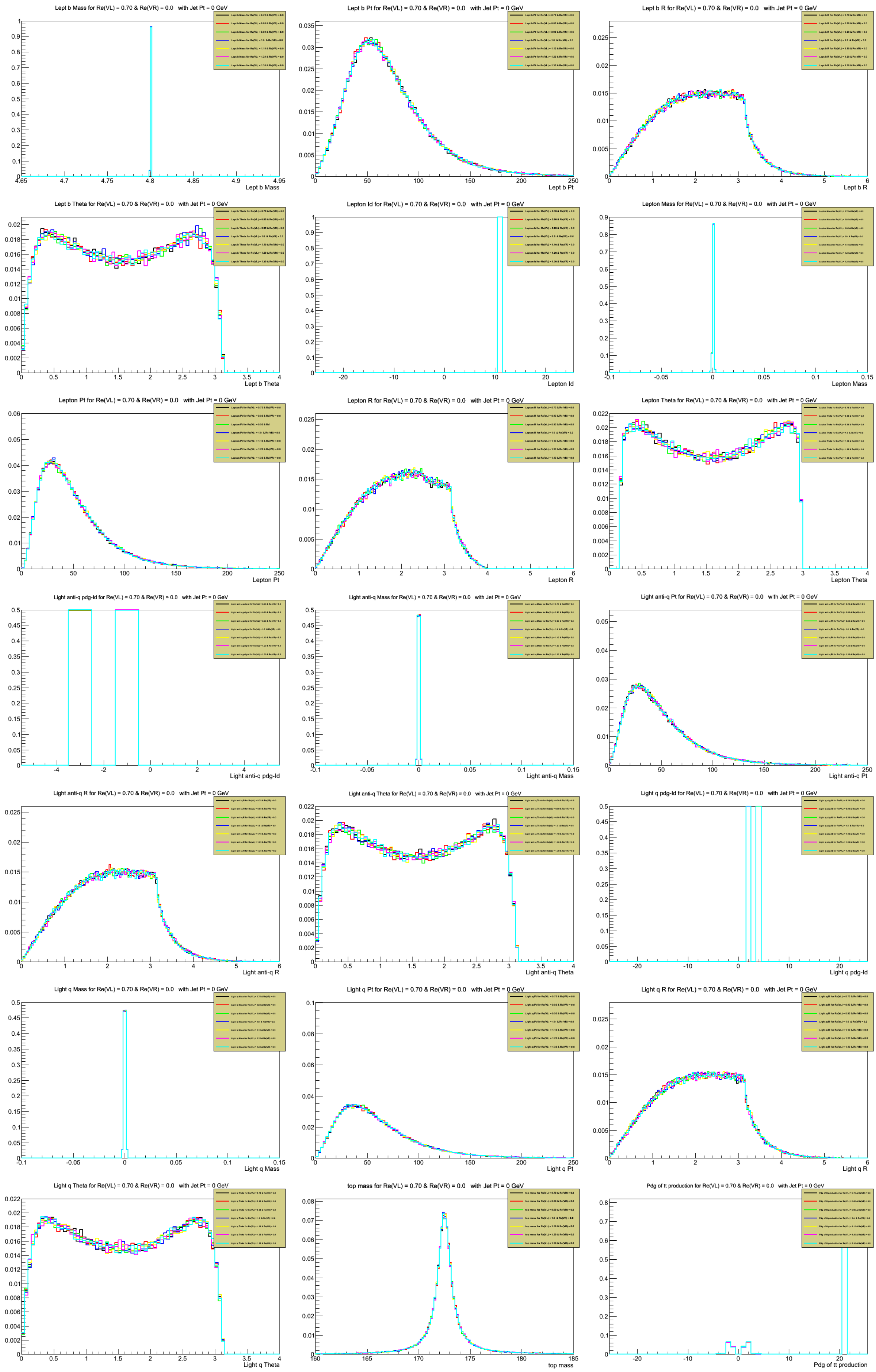


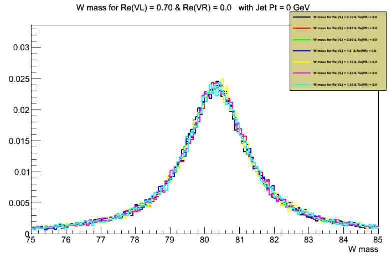


7.2.1.3 Model plots for fixed Pt Cut

The distributions shown in this subsection are for fixed Jet Pt Cut value, set to 0. Also no Pt cut on the lepton was applied. In this case the real part of the left-handed vector coupling has been varied between 0.7 and 1.3 in steps of 0.1 while the right-handed parameter has been fixed to its Standard Model value (0.0).




Lept b Mass for Re(VL) = 0.70 & Re(VR) = 0.0 with Jet Pt = 0 GeV
Lept b Mass
Lept b Pt for Re(VL) = 0.70 & Re(VR) = 0.0 with Jet Pt = 0 GeV
Lept b Pt
Lept b R for Re(VL) = 0.70 & Re(VR) = 0.0 with Jet Pt = 0 GeV
Lept b R
Lept b Theta for Re(VL) = 0.70 & Re(VR) = 0.0 with Jet Pt = 0 GeV
Lept b Theta
Lepton Id for Re(VL) = 0.70 & Re(VR) = 0.0 with Jet Pt = 0 GeV
Lepton Id
Lepton Mass for Re(VL) = 0.70 & Re(VR) = 0.0 with Jet Pt = 0 GeV
Lepton Mass
Lepton Pt for Re(VL) = 0.70 & Re(VR) = 0.0 with Jet Pt = 0 GeV
Lepton Pt
Lepton R for Re(VL) = 0.70 & Re(VR) = 0.0 with Jet Pt = 0 GeV
Lepton R
Lepton Theta for Re(VL) = 0.70 & Re(VR) = 0.0 with Jet Pt = 0 GeV
Lepton Theta
Light anti-q pdg-Id for Re(VL) = 0.70 & Re(VR) = 0.0 with Jet Pt = 0 GeV
Light anti-q pdg-Id
Light anti-q Mass for Re(VL) = 0.70 & Re(VR) = 0.0 with Jet Pt = 0 GeV
Light anti-q Mass
Light anti-q Pt for Re(VL) = 0.70 & Re(VR) = 0.0 with Jet Pt = 0 GeV
Light anti-q Pt
Light anti-q R for Re(VL) = 0.70 & Re(VR) = 0.0 with Jet Pt = 0 GeV
Light anti-q R
Light anti-q Theta for Re(VL) = 0.70 & Re(VR) = 0.0 with Jet Pt = 0 GeV
Light anti-q Theta
Light q pdg-Id for Re(VL) = 0.70 & Re(VR) = 0.0 with Jet Pt = 0 GeV
Light q pdg-Id
Light q Mass for Re(VL) = 0.70 & Re(VR) = 0.0 with Jet Pt = 0 GeV
Light q Mass
Light q Pt for Re(VL) = 0.70 & Re(VR) = 0.0 with Jet Pt = 0 GeV
Light q Pt
Light q R for Re(VL) = 0.70 & Re(VR) = 0.0 with Jet Pt = 0 GeV
Light q R
Light q Theta for Re(VL) = 0.70 & Re(VR) = 0.0 with Jet Pt = 0 GeV
Light q Theta
top mass for Re(VL) = 0.70 & Re(VR) = 0.0 with Jet Pt = 0 GeV
top mass
Pdg of tt production for Re(VL) = 0.70 & Re(VR) = 0.0 with Jet Pt = 0 GeV
Pdg of tt production



7.2.1.4 Model plots for varying Pt Cut

Script adapted, but error obtained when running the python script ...
Worked when everything was copied to the TestDir ...
Maybe the use of nohup gives the problem ...

Chapter 8

MadGraph/MadWeight issues

8.1 Discussion with Olivier Mattelaer (8-9/01/2015)

During this discussion moment with Olivier Mattelaer many interesting subjects have been discussed, the most important of them are summarized below. A detailed overview with all suggestions and recommendations is also created, but not all of them have been added in this document. Many of these items ended up in the to-do list added further in the section.

8.1.1 Transfer Functions

The suggested way to investigate the influence of the created Transfer Functions is by first running the sample with a pure delta function for parton-level events. Afterwards the kinematics of the parton-level events of this sample should be smeared with the created Transfer Functions and then the sample should be ran again, this time with the new Transfer Functions. This ensures that in both cases the Transfer Function which is applied corresponds to the kinematics of the considered events. Any difference between the two results indicates a bias introduced by the Transfer Function.

An important remark was the missing normalisation of the double Gaussian Transfer Function. Currently the necessary normalisation factors have not been added and should be done as soon as possible. This because a non-normalized Transfer Function could result in wrong weights and introduce a significant bias to the obtained results.

A final important comment about the correct implementation of personal Transfer Functions is the behaviour outside the considered fit range. The strange distributions obtained for p_T values outside the considered fit range did not appear in the results of Olivier (and Arnaud) because they had a different fitting algorithm which explicitely forced the Minuit fitter to search only for a local minimum where the relative normalisation of both Gaussians was positive[1].
Since probabilities should always remain positive, the negative distributions should be excluded from the allowed options. This can be done by either adapting the used fitting algorithm to only consider positive probabilities or by simply changing the fit formula to use $max(0, a_3)$ in stead of a_3. The second solution is the most straightforward to

[1]This is actually an allowed and physically motivated restriction which can be applied since it simply corresponds to requiring a probability to always stay positive.

implement, but has as disadvantage that any event with very high or low p_T will have weight 0 and hence be thrown away.

8.1.2 Likelihood normalization

The correct formula for the $\ln(\mathcal{L})$ is given in Equation (8.1):

$$-\sum_{i}^{N} \ln(\frac{1}{\sigma Acc} weight_{MW}) = N \ln \sigma + N \ln Acc - \sum_{i}^{N} \ln weight_{MW} \tag{8.1}$$

The above given normalisation formula has already been applied succesfully on 1D-variation of the V_R parameter while keeping the V_L value fixed to its Standard Model expectation value. However similar success has not been obtained for the variation of the V_L component. It is currently been investigated whether this is caused by considering a too restricted grid or whether it is caused by a lack of sensitivity of the V_L variable. Both possibilities are likely because the observed variation between the different V_L configurations is comparable to the uncertainty introduced by the XS normalisation part in the formula.

This directly points the largest danger of using such a normalisation based on the XS of the considered configuration, namely the strong dependence on the uncertainty of the calculated cross section. This is illustrated in Equation (8.2):

$$unc \sim N \frac{\Delta\sigma}{\sigma} + \sum \frac{\Delta weight_{MW}}{weight_{MW}} \tag{8.2}$$

The first term in this Equation scales as N while the second term only scales as $\sqrt{N}$ meaning that the first term quickly starts to dominate the total uncertainty. Therefore it is important to calculate these MadGraph cross sections for a large number of events in order to reduce the uncertainty as much as possible. The size of the relative uncertainty is also closely related to the number of events which will be calculated by MadWeight and hence a good balance should be found[2].

8.1.3 Cluster optimization

When discussing with Olivier about the restrictions of the IIHE cluster he mentioned the possibility to set a maximum number of jobs which can be submitted simultaneously to the cluster. This means that this maximum can be set to 2000 in order to be in agreement with the maximum number of allowed jobs at the IIHE cluster. There exists even the option to explicitely set the number of remaining jobs which should be reached before a new bunch of jobs should be submitted. Hence this allows to wait until the number of jobs running is reduced to about 1500 before a new set of jobs is submitted in order to reach again the maximum number of 2000.
This can all be changed in the *cluster.py* file which has a class “PBSCluster” at about line 1016.

[2]Important to note is that the N in the uncertainty formula should not be the same as the number of events used to calculate the MadGraph cross section. This N variable is really the number of events which are submitted to MadWeight in order to calculate the corresponding weight for each specific configuration.

The huge benefit of the above mentioned cluster optimization is that it allows to keep using the "collect" option of MadWeight without the necessity of combining multiple separate runs of each 2000 jobs. Hence one should just wait until all the events have been submitted and run the final collect step of the MadWeight setup.

8.1.4 Optimizations and bug fix

During this two-day discussion moment a couple of small optimizations of the used MadWeight configuration have been discovered.
At first it was clear that the used MadWeight version was considered to be an old version for Olivier. Since rather recently a stable version is kept with multiple updates and necessary bug-fixes. The installation command for this newer version is the following and is currently installed as a new directory $NewestMW\_amcnlo$ on $/localgrid$.

$$bzr \quad branch \quad lp : mg5amcnlo \tag{8.3}$$

Secondly the issue of negative weights when using a δ Transfer Function was resolved. This was caused by a discrepancy between the integral implemented in MadWeight and the one developed by Olivier. The reason for a second integral is in order to deal with multiple Transfer Functions during one single submission. The bug-fix was directly created by Olivier and is currently implemented in all directories on $/localgrid$.

A final but rather important remark was the fact that the developed model allowed CKM suppressed W-boson decays. This did not influence the final result since the probability for such decays was extremely small but significantly enlarged the CPU time since MadWeight considers each possible decay and does the calculation of the probability for all of them. Hence excluding these types of decays should improve the CPU running time with possible a factor 4.
The restrictions on the model can easily be added in MadGraph as explained in the launchpad FAQ, and are currently implemented in the newest */localgrid* directory.

Chapter 9

Event Selection

Chapter 10

Event corrections and reconstruction