TECHNISCHE UNIVERSITÄT DRESDEN

A Study of Monte Carlo Methods and their Application to Diphoton Production at the Large Hadron Collider

Bachelor-Arbeit
zur Erlangung des Hochschulgrades
Bachelor of Science
im Bachelor-Studiengang Physik

vorgelegt von

Valentin Boettcher
geboren am 06.06.1998 in Gera (Thüringen)

Institut für Kern- und Teilchenphysik
Fachrichtung Physik
Fakultät Mathematik und Naturwissenschaften
Technische Universität Dresden
2020

Eingereicht: 10. Juni 2020

1. Gutachter: Dr. Frank Siegert
2. Gutachter: Prof. Dr. Michael Kobel

Abstract

The analytical cross section for the parton-level $q\bar{q} \to \gamma\gamma$ diphoton process is calculated in leading order. With the application to this process in mind, some Monte Carlo methods for integration and sampling are studied, implemented and compared. The hard diphoton process in Proton-Proton scattering is simulated in leading order using parton density functions. Throughout, the obtained results are verified through comparison with the `Sherpa` event generator. Finally, a more holistic phenomenological study is performed on the diphoton process by simulating and comparing additional effects including parton showers, hadronization and multiple interactions at *LHC* conditions with the `Sherpa` event generator.

Zusammenfassung

Der analytische Wirkungsquerschnitt für den $q\bar{q} \to \gamma\gamma$ diphoton Prozess wird in führender Ordnung berechnet. In Hinblick auf die Anwendung auf diesen Prozess werden Monte Carlo Methoden für Integration und zur Generierung von Stichproben aus Warscheinlichkeitsverteilungen untersucht, implementiert und verglichen. Der harte diphoton Prozess wird in der Proton-Proton Streuung mithilfe von Partondichtefunktionen in führender Ordnung simuliert. Gewonnene Resultate werden durchweg mit dem `Sherpa` Event Generator verifiziert. Zuletzt wird der diphoton Prozess mit dem `Sherpa` Event Generator in einer realistischeren Simulation untersucht, die unter anderem Parton Shower, Hadronisierung und Multiple Interactions berücksichtigt und deren Einfluesse vergleicht.

Contents

1 Introduction

Monte Carlo (MC) methods have been and still are one of the most important tools for numerical calculations in particle physics. Be it for validating the well established Standard Model (SM) or for making predictions about new theories, MC simulations are the crucial interface of theory and experimental data, making them directly comparable. In this thesis, the use of MC methods will be traced through from simple integration to the simulation of proton-proton scattering.

The test subject here is the quark-antiquark annihilation into two photons $q\bar{q} \to \gamma\gamma$, henceforth called the diphoton process. It forms an important background to the Higgs decay channel $H \to \gamma\gamma$, which was instrumental in its discovery [1, 2], and to a dihiggs decay $HH \to b\bar{b}\gamma\gamma$, a process of recent interest [3] to study the higgs self coupling and probe the limits of the SM. All the while, the diphoton process is still pure QED (Quantum Electrodynamics) at leading order and thus calculable by hand within the scope of this thesis as is being done in Chapter 2. The obtained result is compared to the total cross section obtained with the `Sherpa` [4] event generator, used as matrix element integrator. In Chapter 3 some simple MC methods are discussed, implemented and their results compared. After studying some basic MC integration methods, the `VEGAS` algorithm [5] is implemented and evaluated. Subsequently MC sampling methods, which are closely related to the integration methods, are explored and the output of `VEGAS` is used to improve the sampling efficiency. Histograms of observables are generated and compared to histograms from `Sherpa` using the `Rivet` [6] analysis framework. Chapter 4 deals with proton-proton scattering in the partonic picture using parton density functions, ending with the implementation of a simple event generator for $pp \to \gamma\gamma$ scattering at *LHC* conditions. Some integration and sampling algorithms and their implementations are adapted to the multidimensional case and histograms of observables are generated with good efficiency. Because a real pp scattering event also entails processes like parton showers, hadronization and multiple interactions, a realistic simulation must account for those effects. The impact of those effects on observables is studied in Chapter 5 using the `Sherpa` event generator.

1.1 Conventions

Throughout, natural units with $c = 1, \hbar = 1, k_B = 1, \varepsilon_0 = 1$ are used unless stated otherwise. The fine structure constant's value $\alpha = 1/137.036$ is configured in `Sherpa` and used in analytic calculations.

1.2 Source Code

The (literate) `Python` code, used to generate most of the results and figures can be found under `https://github.com/vale981/bachelor_thesis/` and more specifically in the subdirectory `prog/python/qqgg`.

The file `monte_carlo.py` implements all the Monte Carlo algorithm related functionality as a module. The file `analytical_xs.org` contains a literate computation notebook that generates all the results of Chapter 3. The file `parton_density_function_stuff.org` contains all the computations for Chapter 4. The `Python` code makes heavy use of scipy [7] and its component numpy).

1.3 Software Versions

In Section 2.2 and Chapter 3 the development version of `Sherpa` has been used. Chapters 4 and 5 use version `2.2.10` for reasons of stability.

In the whole thesis, the version `3.1.0` of `Rivet` was used.

2 The Diphoton Process

Consider the scattering of quark and antiquark into two photons $q\bar{q} \to \gamma\gamma$, here called the diphoton process. In leading order this process is being described by the Feynman diagrams in Fig. 2.2. Because there is only QED involved, the color degrees of freedom average out and will not be considered henceforth. Furthermore a high energy regime will be supposed and therefore masses will be neglected.

2.1 Calculation of the Cross Section to Leading Order

After labeling the incoming quarks and outcoming photons, as well as the momenta according to Fig. 2.2, the Feynman rules for QED yield the matrix elements in (2.1), where Z is the electric charge number of the quark and g is the QED coupling constant. The respective spinors and polarization vectors are u, v and ε. Parenthesis are being used, whenever indices would clutter the notation. The matrix element for Fig. 2.2b is obtained by simply renaming $3 \leftrightarrow 4$.

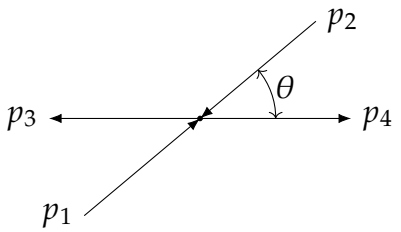


Figure 2.1: Momentum diagram for the process $q\bar{q} \to \gamma\gamma$ in the massles limit.

$$\mathcal{M}_1 = \frac{(gZ)^2}{(p_1 - p_4)^2}\, \bar{\mathrm{v}}(1)\, \not{\varepsilon}^*(4)(\not{p}_1 - \not{p}_4)\, \not{\varepsilon}^*(3)\, \mathrm{u}(2) \tag{2.1}$$

$$\mathcal{M}_2 = \frac{(gZ)^2}{(p_1 - p_3)^2}\, \bar{\mathrm{v}}(1)\, \not{\varepsilon}^*(3)(\not{p}_1 - \not{p}_3)\, \not{\varepsilon}^*(4)\, \mathrm{u}(2) \tag{2.2}$$

a) u channel b) t channel

Figure 2.2: Leading order diagrams for $q\bar{q} \to \gamma\gamma$.

To simplify notation, some shorthands are introduced in (2.3).

$$\begin{aligned} s(x) &= \sin(x) \qquad c(x) = \cos(x) \\ s'(x) &= \sin\left(\frac{x}{2}\right) \quad c'(x) = \cos\left(\frac{x}{2}\right) \end{aligned} \tag{2.3}$$

All calculations are made with respect to the center of momentum (c.m.) frame of the initial state quarks unless stated otherwise. The momenta in the c.m. frame are concretized in (2.4) as illustrated in Fig. 2.1. Note that the photons are aligned to the z-axis as this led to simpler polarization vectors, when calculating the matrix element directly. Here casimir's trick will be used but the labeling was kept.

$$p_1 = p \cdot \begin{pmatrix} 1 \\ s \\ 0 \\ c \end{pmatrix} \qquad p_2 = p \cdot \begin{pmatrix} 1 \\ -s \\ 0 \\ -c \end{pmatrix} \qquad p_3 = p \cdot \begin{pmatrix} 1 \\ 0 \\ 0 \\ -1 \end{pmatrix} \qquad p_4 = p \cdot \begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \end{pmatrix} \tag{2.4}$$

Now observe that $(p_1 - p_4)^2 = -4p^2 s'^2$ and $(p_1 - p_3)^2 = -4p^2 c'^2$ (Minkowski metric) and define Γ_1 and Γ_1 as in (2.5).

$$\Gamma_1 = \not{\epsilon}^*(4)(\not{p}_1 - \not{p}_4)\,\not{\epsilon}^*(3) \qquad \Gamma_2 = \not{\epsilon}^*(3)(\not{p}_1 - \not{p}_3)\,\not{\epsilon}^*(4) \tag{2.5}$$

The total matrix element (the minus sign has been dropped) is given in (2.6).

$$\mathcal{M} = \mathcal{M}_1 + \mathcal{M}_2 = \frac{(gZ)^2}{(2p)^2}\,\bar{\mathrm{v}}(1)\left(\frac{\Gamma_1}{s'^2} + \frac{\Gamma_2}{c'^2}\right)\mathrm{u}(2) \tag{2.6}$$

To obtain an experimentally verifiable cross section the absolute square of the matrix element has to be averaged over incoming helicities and summed over all photon polarizations. Using casimir's trick, the averaging can be simplified to the calculation of a trace as in (2.7) where s_i are helicities, λ_i are the polarizations and $\bar{\Gamma}_i = \gamma^0 \Gamma_i^\dagger \gamma^0$. An additional factor of $\frac{1}{3}$ arises from the color averaging. There are total of *nine* color combinations, but only *three* contribute[1].

$$\langle |\mathcal{M}|^2 \rangle = \frac{1}{3}\frac{1}{4}\sum_{s_1 s_2}\sum_{\lambda_1 \lambda_2} |\mathcal{M}|^2 = \overbrace{\frac{1}{3}\frac{1}{4}\frac{(gZ)^4}{(2p)^4}}^{\mathfrak{F}} \sum_{\lambda_1 \lambda_2} \mathrm{tr}\left[\left(\frac{\Gamma_1}{s'^2} + \frac{\Gamma_2}{c'^2}\right)\not{p}_2\left(\frac{\bar{\Gamma}_1}{s'^2} + \frac{\bar{\Gamma}_2}{c'^2}\right)\not{p}_1\right] \tag{2.7}$$

With the definition $a_1 = 4, b_1 = 3, a_2 = 3, b_2 = 4$ the Γ matrices and their conjugated variants can be written as in (2.8).

$$\Gamma_i = \not{\epsilon}^*(a_i)(\not{p}_1 - \not{p}(a_i))\,\not{\epsilon}^*(b_i) \qquad \bar{\Gamma}_i = \not{\epsilon}(b_i)(\not{p}_1 - \not{p}(a_i))\,\not{\epsilon}(a_i) \tag{2.8}$$

[1] Different color states are orthogonal and there are no preferred colors in the beams.

To work out (2.7) one must evaluate terms of the form (2.9).

$$\begin{aligned}\Gamma_{ij} &= \sum_{\lambda_1\lambda_2} \mathrm{tr}\big(\Gamma_i \not{p}_2 \bar{\Gamma}_j \not{p}_1\big) \\ &= \sum_{\lambda_1\lambda_2} \mathrm{tr}\big[\not{\epsilon}^*(a_i)(\not{p}_1 - \not{p}(a_i))\, \not{\epsilon}^*(b_i)\, \not{p}_2\, \not{\epsilon}(b_j)(\not{p}_1 - \not{p}(a_j))\, \not{\epsilon}(a_j)\, \not{p}_1\big]\end{aligned} \tag{2.9}$$

The sum over polarization can be simplified by utilizing the completeness relation for polarization vectors for *real* photons (2.10).

$$\sum_{\lambda=1}^{2} \varepsilon^{\mu}_{(\lambda)}\, \varepsilon^{*\nu}_{(\lambda)} = -g^{\mu\nu} \tag{2.10}$$

For $i = j$ (2.11) follows by utilizing $\gamma_\mu \not{a} \gamma^\mu = -2\not{a}$, $\gamma_\mu \not{a} \not{b} \not{c} \gamma^\mu = -2\not{c}\not{b}\not{a}$ as well as $\not{p}_i \not{p}_i = p_i \cdot p_i = 0$ and the well known trace theorems for the gamma matrices.

$$\begin{aligned}\Gamma_{ii} &= \mathrm{tr}\big[\gamma_\mu(\not{p}_1 - \not{p}(a_i))\gamma_\nu\, \not{p}_2\, \gamma^\nu(\not{p}_1 - \not{p}(a_i))\gamma^\mu\, \not{p}_1)\big] \\ &= 4\,\mathrm{tr}\big[(\not{p}_1 - \not{p}(a_i))\, \not{p}_2(\not{p}_1 - \not{p}(a_i))\, \not{p}_1\big] \\ &= 32\big[(p(a_i)\cdot p_2)(p(a_i)\cdot p_1)\big]\end{aligned} \tag{2.11}$$

The same tricks as well as the commutation relation for gamma matrices can be utilized for the case $i \neq j$ and lead to (2.12), albeit with more technical effort.

$$\begin{aligned}\Gamma_{ij} &= \mathrm{tr}\big[\gamma_\mu(\not{p}_1 - \not{p}(a_i))\gamma^\nu\, \not{p}_2\, \gamma^\mu(\not{p}_1 - \not{p}(a_j))\gamma_\nu\, \not{p}_1)\big] \\ &= -2\,\mathrm{tr}\big[\not{p}_2\, \gamma^\nu(\not{p}_1 - \not{p}(a_i))(\not{p}_1 - \not{p}(a_j))\gamma_\nu\, \not{p}_1)\big] \\ &= -16\big[(p_1\cdot p_2)(2(p_a\cdot p(a_j)) - p(a_i)\cdot p(a_j)) + (p_1\cdot p(a_i))(p_2\cdot p(a_j)) - (p_1\cdot p(a_j))(p_2\cdot p(a_i))\big]\end{aligned} \tag{2.12}$$

The crucial step here was to sum over μ and utilizing $\gamma^\mu\gamma^\nu\gamma^\rho\gamma^\sigma\gamma_\mu = -2\gamma^\sigma\gamma^\rho\gamma^\nu$.

Multiplying out the terms in (2.7) and plugging in (2.11) and (2.12) results in the averaged matrix element of (2.13). It is noteworthy that the mixing terms cancel out, in other terms: $\Gamma_{12} + \Gamma_{21} = 0$. The result can also be expressed in terms of the pseudo-rapidity $\eta \equiv -\ln[\tan\big(\frac{\theta}{2}\big)]$.

$$\begin{aligned}\langle|\mathcal{M}|^2\rangle &= p^4 \cdot \mathfrak{F} \cdot 32 \cdot \left[\frac{(1-c)(1+c)}{s'^4}\right] + \left[\frac{(1-c)(1+c)}{c'^4}\right] \\ &= \frac{4}{3}(gZ)^4 \cdot \frac{1+\cos^2(\theta)}{\sin^2(\theta)} = \frac{4}{3}(gZ)^4 \cdot (\tanh\big(\eta\big)^2 + 1)\end{aligned} \tag{2.13}$$

2.2 Discussion and Comparison with Sherpa

The result obtained in Section 2.1 shall now be verified by the MC event generator `Sherpa` [4]. To facilitate this, an expression for the total cross section for a range of θ or η has to be obtained. Using Fermi's golden rule for $2 \to 2$ processes and observing that the initial and

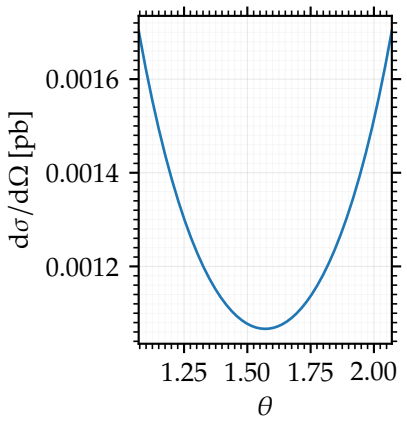
final momenta are equal ($p_i = p_f$) and $g = \sqrt{4\pi\alpha}$, the result (2.14) arises. The differential cross section has also been calculated in terms of the pseudo-rapidity in (2.15).

An additional factor of $\frac{1}{2}$ comes in due to there being two identical photons in the final state.

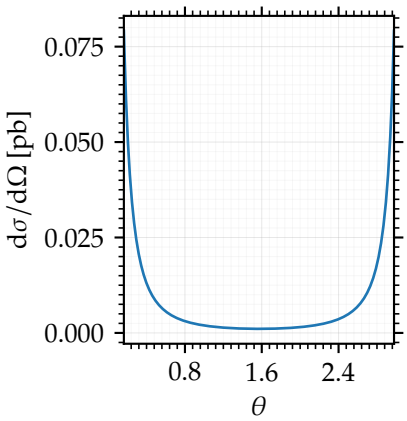
$$\frac{\mathrm{d}\sigma}{\mathrm{d}\Omega} = \frac{1}{2}\frac{1}{(8\pi)^2}\cdot\frac{|\mathcal{M}|^2}{\mathrm{E_{CM}}^2}\cdot\frac{|p_f|}{|p_i|} = \underbrace{\frac{\alpha^2 Z^4}{6\,\mathrm{E_{CM}}^2}}_{\mathfrak{C}}\frac{1+\cos^2(\theta)}{\sin^2(\theta)} \tag{2.14}$$

$$\frac{\mathrm{d}\sigma}{\mathrm{d}\eta} = 2\pi\cdot\frac{\alpha^2 Z^4}{6\,\mathrm{E_{CM}}^2}\cdot\left(\tanh(\eta)^2+1\right) \tag{2.15}$$

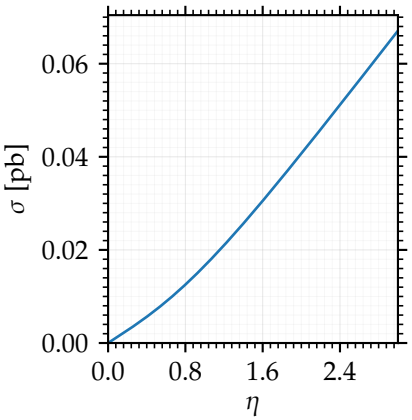
The differential cross section (2.14) (see also Fig. 2.3b) is divergent for angles near zero or π,



a) The differential cross section as a function of the polar angle θ in the crucial region.



b) The differential cross section as a function of the polar angle θ over the full integration interval.



c) The cross section (2.16) of the process for a pseudo-rapidity integrated over $[-\eta, \eta]$.

Figure 2.3: Plots of the differential and total cross section for $q\bar{q} \to \gamma\gamma$.

as expected for massless t- or u-channel diagrams, but remains finite in the physical region

(see Fig. 2.3a). At small scattering angles the absolute square of the momentum carried by the virtual quark in Fig. 2.2 goes to zero, which in the mass-less limit means that the virtual quark comes *on-shell* and thus the process becomes resonant. The divergence of the cross section itself is also not the problem here, because it can be transformed into a form where the divergence does not occur (see (2.15)).

The differential cross section clearly is symmetric around $\theta = \frac{\pi}{2}$ as was to be expected, because the photons are indistinguishable. To compare the cross section to experiment and to simulation an interval around $\theta = \frac{\pi}{2}$ has to be chosen, where the leading order, mass-less approximation may yield a physical result.

The total cross section in such an interval, given by integrating (2.14) for $\theta \in [\theta_1, \theta_2]$ or $\eta \in [\eta_1, \eta_2]$ is given in (2.16).

$$\begin{aligned}\sigma &= 2\pi\mathfrak{C} \cdot \{\cos(\theta_2) - \cos(\theta_1) + 2[\operatorname{artanh}(\cos(\theta_1)) - \operatorname{artanh}(\cos(\theta_2))]\} \\ &= 2\pi\mathfrak{C} \cdot \left[\tanh(\eta_2) - \tanh(\eta_1) + 2(\eta_1 - \eta_2)\right] \\ &= \frac{\pi\alpha^2 Z^4}{3\,\mathrm{E_{CM}}^2} \cdot \left[\tanh(\eta_2) - \tanh(\eta_1) + 2(\eta_1 - \eta_2)\right]\end{aligned} \tag{2.16}$$

As can be seen in Fig. 2.3c, the cross section, integrated over an interval of $[-\eta, \eta]$, is dominated by the linear contributions in (2.16) and would result in an infinity if no cut on η would be imposed. Choosing $|\eta| \leq 2.5$ and $\mathrm{E_{CM}} = 200\,\mathrm{GeV}$ the process was MC integrated in `Sherpa` using the runcard in Appendix A.1.1. This runcard describes the exact same (leading order) process as the calculated cross section.

`Sherpa` arrives at the cross section $\sigma = (0.053\,793\,8 \pm 0.000\,002\,5)\,\mathrm{pb}$. Plugging the same parameters into (2.16) gives $\sigma = 0.053\,793\,3\,\mathrm{pb}$ which is within the uncertainty range of the `Sherpa` value. This verifies the result for the total cross section.

3 Survey of Elementary Monte Carlo Methods

Monte Carlo methods for multidimensional integration and sampling of probability distributions are central tools of modern particle physics. Therefore some simple methods and algorithms are being studied and implemented from scratch here and will be applied to the results from Chapter 2. The `Python` code for the implementation can be found as described in Section 1.2. The sampling and integration intervals, as well as other parameters have been chosen as in Section 2.2, so that $|\eta| \leq 2.5$ and $E_{CM} = 200\,\text{GeV}$. This chapter is based on the appendix of [8] and supplants that source with some derivations and more methods, for instance the `VEGAS` [5] algorithm.

3.1 Monte Carlo Integration

Consider a function $f : \mathbf{x} \in \Omega \subset \mathbb{R}^n \mapsto \mathbb{R}$ and a probability density on $\rho : \mathbf{x} \in \Omega \mapsto \mathbb{R}_{\geq 0}$ with $\rho(\Omega) > 0$, $\int_\Omega \rho(\mathbf{x})\,\mathrm{d}\mathbf{x} = 1$. By multiplying f with a one in the fashion of (3.1), the integral of f over Ω can be interpreted as the expected value $\mathrm{E}[F/P]$ of the random variable F/P under the distribution ρ. This is the key to most MC methods.

$$I = \int_\Omega f(\mathbf{x})\,\mathrm{d}\mathbf{x} = \int_\Omega \left[\frac{f(\mathbf{x})}{\rho(\mathbf{x})}\right] \rho(\mathbf{x})\,\mathrm{d}\mathbf{x} = \mathrm{E}\left[\frac{F}{P}\right] \tag{3.1}$$

The expected value $\mathrm{E}[F/P]$ can be approximated by calculating the mean of F/P with N finite samples $\{\mathbf{x}_i\}_{i\in\overline{1,N}} \sim \rho$ (distributed according to ρ), where N is a very large integer.

$$\mathrm{E}\left[\frac{F}{P}\right] \approx \frac{1}{N}\sum_{i=1}^{N} \frac{f(\mathbf{x_i})}{\rho(\mathbf{x_i})} \xrightarrow{N\to\infty} I \tag{3.2}$$

The convergence of (3.2) is due to the nature of the expected value (3.3) and variance (3.4) of the mean $\overline{X} = \frac{1}{N}\sum_i X_i$ of N uncorrelated random variables $\{X_i\}_{i\in\overline{1,N}}$ with the same distribution, expected value $\mathrm{E}[X_i] = \mathbb{E}$ and variance $\sigma_i^2 = \sigma^2$.

$$\mathrm{E}\left[\overline{X}\right] = \frac{1}{N}\sum_i \mathrm{E}[X_i] = \mathbb{E} \tag{3.3}$$

$$\sigma_{\overline{X}}^2 = \sum_i \frac{\sigma_i^2}{N^2} = \frac{\sigma^2}{N} \tag{3.4}$$

Because $\frac{\sigma^2}{N} \xrightarrow{N\to\infty} 0$ (3.2) really converges to I. For finite, but large N the value of $\mathrm{E}[F/P]$ is distributed (in increasingly good approximation) according to a normal distribution around I with the variance $\mathrm{VAR}[F/P] \cdot N^{-1}$ as in (3.5) by virtue of the central limit theorem.

$$\mathrm{VAR}\left[\frac{F}{P}\right] = \int_\Omega \left[I - \frac{f(\mathbf{x})}{\rho(\mathbf{x})}\right]^2 \rho(\mathbf{x})\,\mathrm{d}\mathbf{x} = \int_\Omega \left[\left(\frac{f(\mathbf{x})}{\rho(\mathbf{x})}\right)^2 - I^2\right] \rho(\mathbf{x})\,\mathrm{d}\mathbf{x} \tag{3.5}$$

$$\approx \frac{1}{N-1} \sum_i \left[I - \frac{f(\mathbf{x_i})}{\rho(\mathbf{x_i})}\right]^2 \tag{3.6}$$

The goal now is to reduce $\mathrm{VAR}[F/P]$ to speed up the convergence of (3.2) and achieve higher accuracy with fewer function evaluations. There are at least three angles of attack in 3.1, namely the distribution ρ, the variable $\mathbf{x}$, and the integration volume Ω. Accordingly, some ways variance reductions can be accomplished are choosing a suitable ρ (importance sampling), by transforming the integral onto another variable or by subdividing the integration volume into several sub-volumes of different size while keeping the sample size constant in all sub-volumes (a special case of stratified sampling).[1][2] Combining ideas from importance sampling and stratified sampling leads to the `VEGAS` algorithm [5] that approximates the optimal distribution of importance sampling by adaptive subdivision of the integration volume into a grid.

The convergence of (3.2) is independent of the dimensionality of the integration volume as opposed to many other numerical integration algorithms (trapezoid rule, Simpsons rule) that usually converge like $N^{-\frac{k}{n}}$ with $k \in \mathbb{N}_{>0}$ and n being the dimensionality. MC integration in its simplest form converges like $N^{-\frac{1}{2}}$ in *all* dimensions. Because phase space integrals in particle physics usually have a high dimensionality, MC integration is a suitable approach for those problems. When implementing MC methods, the random samples can be obtained through hardware or software random number generators (RNGs). Most implementations utilize software RNGs because they supply pseudo-random numbers in a reproducible way, which facilitates deniability and comparability [8].

3.1.1 Naive Monte Carlo Integration and Change of Variables

The simplest choice for ρ is given by (3.7), the uniform distribution.

$$\rho(\mathbf{x}) = \frac{1}{\int_\Omega 1\,\mathrm{d}\mathbf{x}'} = \frac{1}{|\Omega|} \tag{3.7}$$

With this distribution (3.2) becomes (3.8). In other words, I is just the mean value of f in Ω, henceforth called $\bar{f}$, multiplied with the volume.

$$\mathrm{E}\left[\frac{F}{P}\right] \approx \frac{|\Omega|}{N} \sum_{i=1}^{N} f(\mathbf{x_i}) = |\Omega| \cdot \bar{f} \tag{3.8}$$

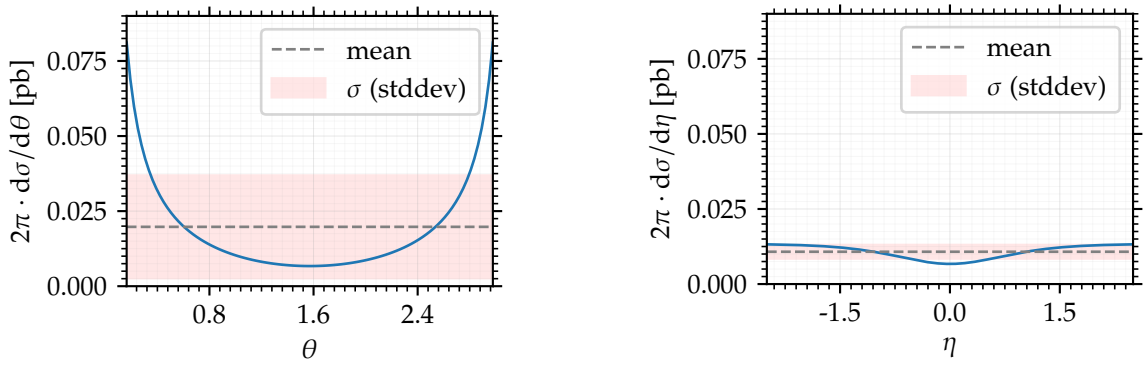
[1] There are of course still other methods like the multi-channel method.
[2] Of course, combinations of these methods can be applied as well.

The variance $\mathrm{VAR}[I] = \mathrm{VAR}[F/P]$ is now given by (3.9). Note that the factor $|\Omega|$ gets squared when approximating the integral by the sum.

$$\mathrm{VAR}\left[\frac{F}{P}\right] = |\Omega| \int_\Omega f(\mathbf{x})^2 - \underbrace{\left(\frac{I}{|\Omega|}\right)^2}_{=\bar{f}} \mathrm{d}\mathbf{x} \equiv |\Omega| \cdot \sigma_f^2 \approx \frac{|\Omega|^2}{N-1} \sum_i \left[f(\mathbf{x}_i) - \bar{f}\right]^2 \tag{3.9}$$

Applying this method to integrate the $q\bar{q} \to \gamma\gamma$ cross section from (2.14) over a θ interval, equivalent to $\eta \in [-2.5, 2.5]$ with a target accuracy of $\sigma = 1 \times 10^{-3}\,\mathrm{pb}$ results in $\sigma = (0.0537 \pm 0.0010)\,\mathrm{pb}$ with a sample size of $N = 2175$.

Changing variables and integrating (2.15) over η with the same target accuracy yields $\sigma = (0.0543 \pm 0.0009)\,\mathrm{pb}$ with a sample size of just $N = 155$. The dramatic reduction in variance and sample size can be understood qualitatively by studying Fig. 3.1, which shows both integrands with the same y-axis scaling and their standard deviation visualized. The differential cross section in terms of η (Fig. 3.1b) is less steep than the differential cross section in terms of θ (Fig. 3.1a) and takes on large values over most of the integration interval. In general, the Jacobian arising in variable transformation has the same effect as the probability density in importance sampling. It can be shown that importance sampling and change of variables are formally equivalent (see A.3.1).



a) The integrand arising from differential cross section $\frac{\mathrm{d}\sigma}{\mathrm{d}\theta}$ with the integration borders visualized as gray lines.

b) The differential cross section $\frac{\mathrm{d}\sigma}{\mathrm{d}\eta}$ (see (2.15)) scaled by 2π with the integration borders visualized as gray lines.

Figure 3.1: Comparison of two parametrisations of the differential cross section. The same y-axis scaling has been chosen to visualize the difference in variance.

3.1.2 Integration with VEGAS

Subdividing the integration volume into sub-volumes and taking the same number of samples in each volume (stratified sampling), gives optimal results, when the variance in every sub-volume is the same [5]. In importance sampling, the optimal probability distribution is given by (3.10), where $f(\Omega) \geq 0$ is presumed without loss of generality. When applying VEGAS

to multi dimensional integrals, (3.10) is usually modified to factorize into distributions for each variable to simplify calculations.

$$\rho(\mathbf{x}) = \frac{f(\mathbf{x})}{\int_\Omega f(\mathbf{y})\,\mathrm{d}y} \tag{3.10}$$

The idea behind `VEGAS` is to subdivide Ω into hypercubes (create a grid), define ρ as step-function with constant value on those hypercubes and iteratively approximating (3.10), instead of trying to minimize the variance directly. As in stratified sampling, the sample number is kept the same in each hypercube to realize (3.10). In the end, the samples are concentrated where f takes on the highest values and changes most rapidly. This is done by subdividing the hypercubes into smaller chunks, based on their contribution to the integral, which is calculated through MC sampling, and then varying the hypercube borders until all hypercubes contain the same number of chunks. So if the contribution of a hypercube is large, it will be divided into more chunks than others. When the hypercube borders are then shifted, this hypercube will shrink, while others will grow by consuming the remaining chunks. Repeating this step in so called `VEGAS` iterations will converge the contribution of each hypercube to the same value. More details about the algorithm can be found in [5]. Note that no knowledge about the integrand is required. The probability density used by `VEGAS` is given in (3.11) with K being the number of hypercubes and Ω_i being the hypercubes themselves.

$$\rho(\mathbf{x}) = K \cdot \begin{cases} \frac{1}{|\Omega_i|} & \mathbf{x} \in \Omega_{i \in \overline{1,K}} \\ 0 & \text{otherwise} \end{cases} \tag{3.11}$$

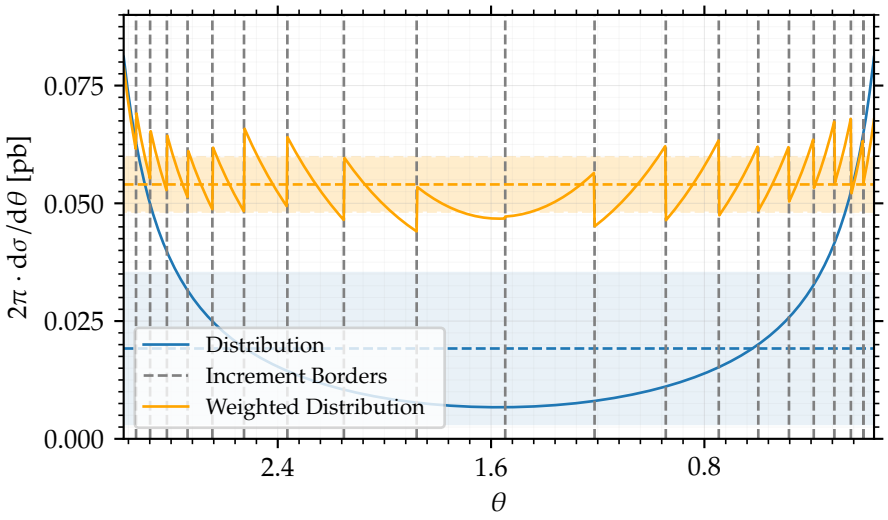


Figure 3.2: The same integrand as in Fig. 3.1a with `VEGAS`-generated increments and weighting applied (f/ρ). The colored bands are the standard deviations of the distributions with matching color.

This algorithm has been implemented in `Python` and applied to (2.14). In one dimension the hypercubes become simple interval increments and applying `VEGAS` to (2.14) with $K = 20$ increments yields $\sigma = (0.0539 \pm 0.0006)\,\mathrm{pb}$ with $N = 280$ function evaluations (including

VEGAS iterations). This result is comparable with the one obtained by parameter transformation in Section 3.1.1. The sample count N is the total number of evaluations of f. The resulting increments and the weighted integrand f/ρ are depicted in Fig. 3.2, along with the original integrand and it is intuitively clear, that the variance is being reduced. Smaller increments correspond to higher sample density and lower weights, flattening out the integrand.

Generally the result gets better with more increments, but at the cost of more VEGAS iterations. The intermediate values of those iterations can be accumulated to improve the accuracy of the end result [5, p. 197].

The VEGAS algorithm can be adapted to n dimensions by using a grid of hypercubes instead of intervals and using the algorithm along each axis with a slightly altered weighting mechanism [5, p. 197].

3.2 Monte Carlo Sampling

Drawing representative samples from a probability distribution (for example a differential cross section) results in a set of *events*. This procedure is called sampling and produces the same kind of data, that is gathered in experiments and from which one can then calculate samples from the distribution of other observables event-by-event without explicit transformation of the distribution. Furthermore, these samples can be used as the basis for more involved simulations (see Chapter 5). Sampling shares many characteristics with integration and thus the methods discussed here use similar terminology and often inherit their fundamental ideas from the integration methods.

Here the one-dimensional case is discussed. Sampling a multi-dimensional distribution is equivalent to sampling one dimensional distributions by reducing the distribution itself to one variable through integration over the remaining variables and then, keeping the first variable fixed, sampling the other variables in a likewise manner.

Consider a function $f\colon x \in \Omega \mapsto \mathbb{R}_{>0}$ where $\Omega = [0,1]$ without loss of generality. Such a function is proportional to a probability density $\tilde{f}$. When X is a uniformly distributed random variable on $[0,1]$ (which can be easily generated), then a sample x_i of this variable can be transformed into a sample of $Y \sim \tilde{f}$. Let x be a single sample of X. A sample y of Y can be obtained by solving (3.12) for y.

$$\int_0^y f(x')\,\mathrm{d}x' = x \cdot \int_0^1 f(x')\,\mathrm{d}x' = x \cdot A \tag{3.12}$$

This can be shown by observing that, according to (3.12), the probability that $y \in [y', y' + \mathrm{d}y']$ is the same as the probability that

$$x \in A^{-1}\left[\int_0^{y'} f(x')\,\mathrm{d}x' \,,\, \int_0^{y'+\mathrm{d}y'} f(x')\,\mathrm{d}x'\right]$$

which is

$$A^{-1}\left(\int_0^{y'+\mathrm{d}y'} f(x')\,\mathrm{d}x' - \int_0^{y'} f(x')\,\mathrm{d}x'\right) = A^{-1} f(y')\,\mathrm{d}y'$$

and therefore y is really distributed according to $f/A = \tilde{f}$. If the antiderivative F (and its inverse) of f is known, then the solution of (3.12) is given by (3.13).

$$y = F^{-1}(x \cdot A + F(0)) \tag{3.13}$$

Note that F is always invertible because F is increasing monotonically. Of course F and its inverse can be obtained numerically or one can change variables to simplify.

3.2.1 Importance Sampling and Hit or Miss

If integrating f or inverting F is too expensive or a fully f-agnostic method is desired, the problem can be reformulated by introducing a positive function $g\colon x \in \Omega \mapsto \mathbb{R}_{\geq 0}$ with $\forall x \in \Omega\colon g(x) \geq f(x)$.

Observing (3.14) suggests, that one generates samples which are distributed according to g/B, where $B = \int_0^1 g(x)\,\mathrm{d}x$ and then accepts them with the probability f/g, so that g cancels out. This is known as the *hit or miss* implementation of importance sampling.

$$\int_0^y f(x')\,\mathrm{d}x' = \int_0^y g(x') \cdot \frac{f(x')}{g(x')}\,\mathrm{d}x' = \int_0^y g(x') \int_0^{\frac{f(x')}{g(x')}} \mathrm{d}z\,\mathrm{d}x' \tag{3.14}$$

The thus obtained samples are then distributed according to f/B and the total probability of accepting a sample, also called the efficiency $\mathfrak{e}$, is given by (3.15).

$$\int_0^1 \frac{g(x)}{B} \cdot \frac{f(x)}{g(x)}\,\mathrm{d}x = \int_0^1 \frac{f(x)}{B}\,\mathrm{d}x = \frac{A}{B} = \mathfrak{e} \leq 1 \tag{3.15}$$

The closer the sizes of volumes enclosed by g and f are to each other, the higher is $\mathfrak{e}$.

Choosing g like (3.16) and looking back at (3.13) yields $y = x \cdot A$, so that the sampling procedure simplifies to choosing random numbers $x \in [0, 1]$ and accepting them with the probability $f(x)/g(x)$. The efficiency of this approach is related to how much f differs from $f_{\max}$ which in turn is related to the variance of f. Minimizing variance will therefore improve sampling performance. The method can also be used in higher dimensions without modification and has again been implemented and evaluated.

$$g = \max_{x \in \Omega} f(x) = f_{\max} \tag{3.16}$$

Using the upper bound defined in (3.16) with the distribution for $\cos\theta$ derived from the differential cross section (2.14) given in (3.17) ($\mathfrak{C}$ being a constant) results in a sampling

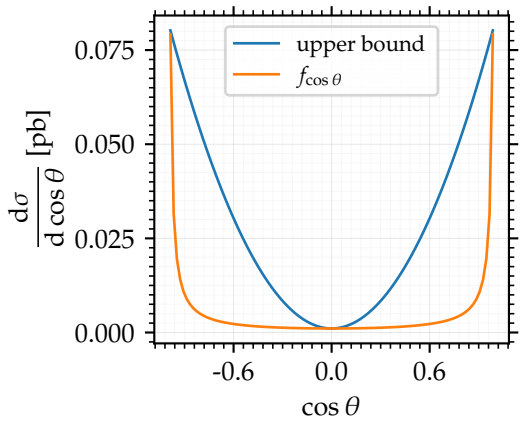


Figure 3.3: The distribution (3.17) and an upper bound of the form $a + b \cdot x^2$.

efficiency of $\mathfrak{e} = 3\%$.

$$f_{\cos\theta}(x = \cos\theta) = \mathfrak{C} \cdot \frac{1 + x^2}{1 - x^2} \tag{3.17}$$

This very low efficiency stems from the fact, that $f_{\cos\theta}$ is a lot smaller than its maximum in most of the sampling interval.

Utilizing an upper bound of the form $a + b \cdot x^2$ with a, b constant improves the efficiency to $\mathfrak{e} = 8\%$. The distribution, as well as the upper bound are depicted in Fig. 3.3.

3.2.2 Importance Sampling through Change of Variables

When transforming f to a new variable $y = y(x)$ one arrives at (3.18) and may reduce variance, analogous to Section 3.1.1. Transforming the distribution in a beneficial way is an alternative method of performing *importance sampling*.

$$\tilde{f} = f(x(y)) \cdot \frac{\mathrm{d}x(y)}{\mathrm{d}y} \tag{3.18}$$

The optimal transformation would be the solution of $y = F(x)$ (F being the antiderivative of f), so that $f(x(y)) \cdot \frac{\mathrm{d}x(y)}{\mathrm{d}y} = 1$. But transforming f in this way is the same as solving (3.12) which is a problem that has been addressed in Section 3.2.1. The difference here is, that we restate the sampling problem in y space, which separates the step of converting our y samples to x samples away from the sampling process (see Section 3.2.5). Solving (3.12) may now be easier, or applying the hit or miss method may be more efficient.

When transforming the differential cross-section to the pseudo rapidity η the efficiency of the hit or miss method rises to $\mathfrak{e} = 41\%$ so applying this method in conjunction with others is worthwhile (see Fig. 3.1).

3.2.3 Hit or Miss and VEGAS

Finding a suitable upper bound or variable transformation requires effort and detail knowledge about the distribution and is hard to automate. Revisiting the idea behind (3.14) but looking at a probability density $\rho > 0$ on Ω leads to a slight reformulation of the method discussed in Section 3.2.1. Note that without loss of generality one can again choose $\Omega = [0, 1]$.

Define $h = \max_{x \in \Omega} f(x)/\rho(x)$, take a sample $\{\tilde{x}_i\} \sim \rho$ distributed according to ρ and accept each sample point with the probability $f(x_i)/(\rho(x_i) \cdot h)$. This is very similar to the procedure described in Section 3.2.1 with $g = \rho \cdot h$, but here the step of generating samples distributed according to ρ is left out as these samples are assumed to be available or can be obtained with little effort (see below). The efficiency of this method is given by (3.19).

$$\mathfrak{e} = \int_0^1 \rho(x) \frac{f(x)}{\rho(x) \cdot h} \, \mathrm{d}x = \frac{A}{h} \tag{3.19}$$

It may seem startling that h determines the efficiency, because h is a (global) maximum and A is an integral but (3.20) states that $\mathfrak{e}$ is well-formed ($\mathfrak{e} \leq 1$). Albeit h is determined through a single point, being the maximum is a global property and there is also the constraint $\int_0^1 \rho(x) \, \mathrm{d}x = 1$ to be considered.

$$A = \int_0^1 \rho(x) \frac{f(x)}{\rho(x)} \, \mathrm{d}x \leq \int_0^1 \rho(x) \cdot h \, \mathrm{d}x = h \tag{3.20}$$

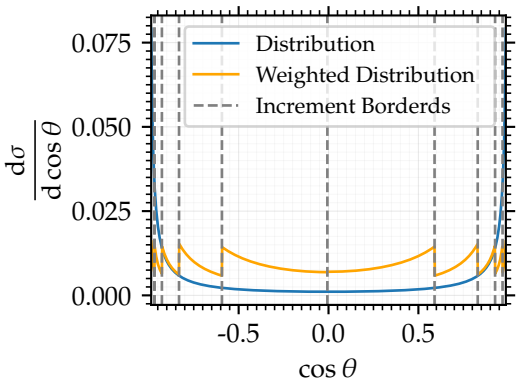
The closer h approaches A the better the efficiency gets. In the optimal case $\rho = f/A$ and thus $h = A$ or $\mathfrak{e} = 1$. This distribution can be approximated in the way discussed in Section 3.1.2 by using the hypercubes found by VEGAS and simply generating the same number of uniformly distributed samples in each hypercube. The distribution ρ takes on the form (3.11). The effect of this approach is visualized in Fig. 3.4a and the resulting sampling efficiency $\mathfrak{e} = 60\%$ (using $K = 10$ increments) is a great improvement over the hit or miss method in Section 3.2.1 and even surpasses the variable transformation to η. By using more increments better efficiencies can be achieved, although the run-time of VEGAS increases. The advantage of VEGAS in this situation is, that the computation of the increments has to be done only once and can be reused. Furthermore, no special knowledge about the distribution f is required.

3.2.4 Stratified Sampling

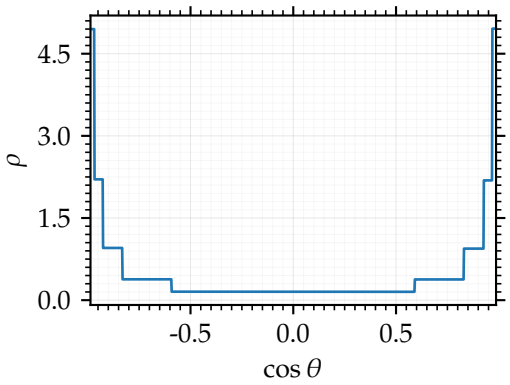
Yet another approach is to subdivide the sampling volume Ω into K sub-volumes $\Omega_i \subset \Omega$ and then take a number of samples from each volume proportional to the integral of the function f in that volume. This is a variant of *stratified sampling*, with the advantage that it is now possible to optimize the sampling in each sub-volume independently.

Let N be the total sample count ($N \gg 1$), $A_i = \int_{\Omega_i} f(x) \, \mathrm{d}x$ and $A = \sum_i A_i$. The total efficiency when taking $N_i = A_i/A \cdot N$ samples in each hypercube is then given by (3.21).

$$\mathfrak{e} = \frac{\sum_i N_i}{\sum_i N_i/\mathfrak{e}_i} = \frac{\sum_i A_i/A \cdot N}{\sum_i A_i/A \cdot N/\mathfrak{e}_i} = \frac{\sum_i A_i}{\sum_i A_i/\mathfrak{e}_i} \tag{3.21}$$



a) The distribution for $\cos\theta$ (see (3.17)) and the VEGAS-weighted distribution.



b) The weighting distribution generated by VEGAS. It is clear, that it closely follows the original distribution (3.17).

Figure 3.4: VEGAS-weighted distribution and weighting distribution.

In the case when all $\mathfrak{e}_i$ are the same, the total efficiency is the same as the individual efficiencies. In the case of one efficiency being much smaller then the others, this efficiency dominates the overall efficiency (assuming somewhat similar A_i). So in general one should optimize so that the individual efficiencies are roughly the same. Using the Ω_i generated by VEGAS has the advantage, that this requirement can be approximated and the A_i have already been obtained by VEGAS. The technical difficulty in the implementation is the way in which sample points get distributed among the sub-volumes. The pitfall here is that the A_i (and the upper bounds for the hit-or-miss method) have to be determined increasingly accurate with growing sample size.

This method will be applied to multi-dimensional sampling in Section 4.3.

3.2.5 Observables

Having obtained a sample of a distribution, distributions of other observables can be calculated from those samples without having to transform the distribution into new variables. This is due to the discrete nature of the samples. Suppose there is an observable $\gamma : \Omega \mapsto \mathbb{R}$. Now to take a sample $\{x_i\}$ of γ we sample f and convert the sample values by simply applying γ, where f is as defined in Section 3.2. This is equivalent to substituting $y = \gamma^{-1}(z)$ in (3.12) and solving for z.

The probability that $z \in [z', z' + \mathrm{d}z']$ now is the same as the probability that

$$x \in A^{-1}\left[\int_0^{\gamma^{-1}(z')} f(x')\,\mathrm{d}x' \,,\, \int_0^{\gamma^{-1}(z')+\partial_z(\gamma^{-1})(z')\mathrm{d}z'} f(x')\,\mathrm{d}x'\right]$$

which is $A^{-1} \cdot f(\gamma^{-1}(z')) \cdot (\partial_z \gamma^{-1})(z')\,\mathrm{d}z'$. That is the same result, as if the distribution had been transformed by multiplying the appropriate Jacobian.

Using the distribution (3.17) for the variable $\cos\theta$ and choosing the polar angle φ uniformly random, a sample of 4-momenta can be generated and histograms of observables can be drawn.

The observables considered here are the transverse momentum p_T and the pseudo rapidity η of one photon which can be computed from the 4-momentum as described in (3.22).

(3.22)

$$p_T = \sqrt{(p_x)^2 + (p_y)^2} \qquad \eta = \operatorname{arcosh}\left(\frac{|\mathbf{p}|}{p_T}\right) \cdot \operatorname{sgn}(p_z)$$

The histograms in Fig. 3.6 have been created by generating $N = 1000000$ samples. Figure 3.6 also contains reference histograms created by generating events with `Sherpa` and analyzing them with the `Rivet` toolkit [6]. The utilized analysis can be found in Appendix A.2.1. Where

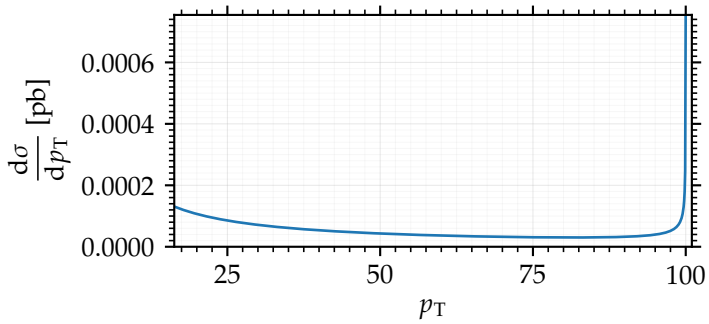
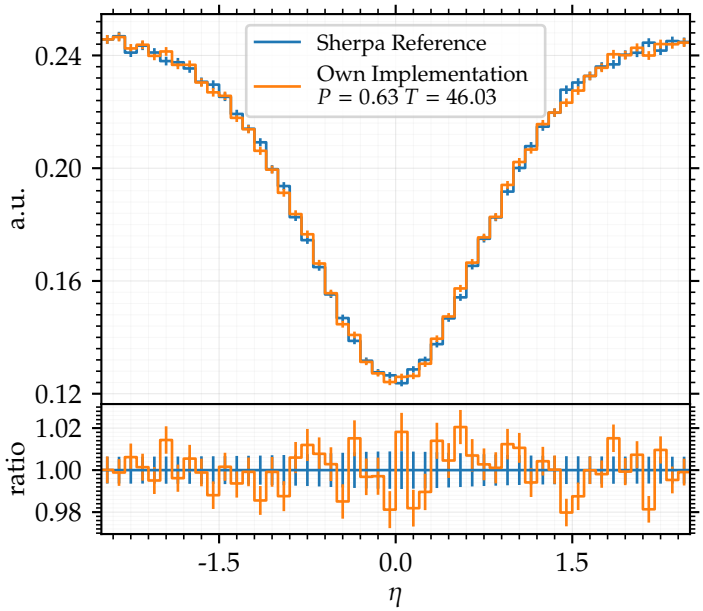


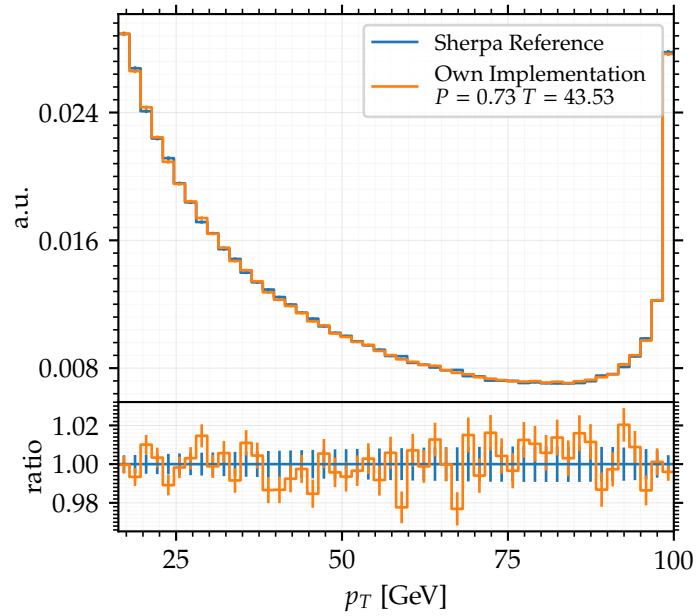
Figure 3.5: The differential cross section transformed to p_T.

Fig. 3.6a shows clear resemblance of Fig. 3.1b, the peak and the rise before this peak in Fig. 3.6b around $p_T = 100\,\text{GeV}$ seems surprising and opens the possibility for the production of hard transverse photons. When transforming the differential cross section to p_T it can be seen in Fig. 3.5 that there really is a singularity at $p_T = E_{CM}$, due to a term $1/\sqrt{1-(2\cdot p_T/E_{CM})^2}$ stemming from the Jacobian determinant. This singularity will vanish once considering a more realistic process (see Chapter 4).

The compatibility of the histograms and their `Rivet`-generated counterparts is tested as is discussed in Appendix A.3.2 and the respective P and T values are being included in the ratio plots. The histograms have a P-value greater than 0.6 and are therefore considered valid.



a) Histogram of the pseudo-rapidity (η) distribution.



b) Histogram of the transverse momentum (p_T) distribution.

Figure 3.6: Histograms of observables, generated from a sample of 4-momenta and normalized to unity. The plots include histograms generated by `Sherpa` and `Rivet`. The P and T values are a measure of consistency of the two histograms as is described in Appendix A.3.2.

4 The Hard Diphoton Process in Proton-Proton Scattering

Because free quarks do not occur in nature, one has to study the scattering of hadrons to obtain experimentally verifiable results. Hadrons are usually modeled as consisting of multiple *partons* (i.e. quarks and gluons) using Parton Density Functions (PDFs). By using a leading order PDF, the cross section for the process $pp \to \gamma\gamma$ on the matrix-element level[1] and event samples of that process are obtained [8, p. 14]. These results are being compared with results from `Sherpa`.

4.1 Parton Density Functions

Parton Density Functions encode, restricting considerations to leading order, the probability to encounter a constituent parton of a hadron with a certain momentum fraction x at a certain factorization scale Q^2 in a scattering process. PDFs are normalized according to (4.1), where the sum runs over all partons.

$$\sum_i \int_0^1 x \cdot f_i(x; Q^2)\, \mathrm{d}x = 1 \tag{4.1}$$

More precisely f_i denotes a PDF set, which is referred to simply as PDF in the following. PDFs can not be derived from first principles and have to be determined experimentally for low Q^2 and can be evolved to higher Q^2 through the *DGLAP* equations [9] at different orders of perturbation theory. In deep inelastic scattering Q^2 is just the negative of the momentum transfer: $-q^2$. For more complicated processes Q^2 has to be chosen in a way that reflects the *energy-momentum scale* of the process. If the perturbation series behind the process would be expanded to the exact solution, the dependence on the factorization scale would vanish. In lower orders, one has to choose the scale in a *physically meaningful* way, which reflects characteristics of the process [9].

In the case of $q\bar{q} \to \gamma\gamma$ the mean of the Mandelstam variables $\hat{t}$ and $\hat{u}$, which is equal to $\hat{s}/2$, can be used. This choice is lorentz-invariant and reflects the t/u-channel nature of the process, although the p_T of photon would also have been a good choice [8, p. 18].

The (differential) hadronic cross section for scattering of two partons in equal hadrons is given in (4.2). Here i, j are the partons participating in a scattering process with the cross section $\hat{\sigma}_{ij}$.

[1] Neglecting the remnants and other processes like parton showers, primordial transverse momentum and multiple interactions.

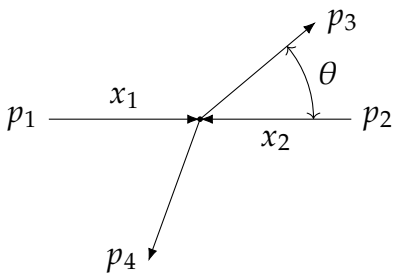
Usually this cross section depends on the kinematics and thus the momentum fractions and the factorization scale[2].

$$\sigma_{ij} = \int f_i(x_1; Q^2)\, f_j(x_2; Q^2)\, \hat{\sigma}_{ij}(x_1, x_2, Q^2)\, \mathrm{d}x_1\, \mathrm{d}x_2 \tag{4.2}$$

Summing (4.2) over all partons in the hadron gives the total scattering cross section for the hadron.

4.2 Kinematics and Cross-Section in the Lab-Frame

To utilize (4.2) for modeling the $pp \to \gamma\gamma$ process and to generate event samples, the results of Chapter 2 have to be transformed into the center of momentum frame of the colliding protons. Quantities in the center of momentum frame of the partons will be starred (like x^*).



Let E_p be the beam energy of the proton beams and x_1, x_2 the momentum fractions of the partons in their respective protons. Choosing the frame where the proton momenta are parallel to the z-Axis and neglecting the mass of the protons, their momenta are $\bar{p}_{1,2} = E_P(1 \quad 0 \quad 0 \quad \pm 1)$. In the limit of high proton energies[3], the quark momenta are simply $p_{1,2} = x_{1,2} \cdot \bar{p}_{1,2}$ and their invariant mass is thus given by (4.3).

$$\mathrm{E_{CM}} = \sqrt{4x_1x_2} \cdot E_P \tag{4.3}$$

The center of momentum frame moves with a velocity $\beta = (x_1 - x_2)/(x_1 + x_2)$. Because the (pseudo-) rapidities are additive under boosts, it is simplest to work with cross section in terms of the pseudo-rapidity. The pseudo-rapidity of a final state photon in the c.m.-frame of the partons is therefore given by (4.4).

$$\eta^* = \eta - w = \eta - \operatorname{artanh}(\beta) \tag{4.4}$$

Because $\mathrm{d}\eta^*/\mathrm{d}\eta = 1$ the cross section from (2.15) becomes (4.5), where η is the pseudo-rapidity one photon.

$$\frac{\mathrm{d}\sigma}{\mathrm{d}\eta} = 2\pi \cdot \frac{\alpha^2 Z^4}{24 E_p^2 x_1 x_2} \cdot \left(\tanh(\eta - w)^2 + 1\right) \tag{4.5}$$

The 4-momenta of the final state photons can be obtained by explicit Lorentz transformation and are listed in (4.6) where θ is the polar angle of the "first" final state photon. These relations

[2] More appropriately: The factorization scale depends on the process. So $\sigma(Q^2)$ is just a symbol for that relation.
[3] When the transverse momentum of the partons can be neglected.

are easily transcribed to η dependence by using the identity $\cos(\theta) = \tanh(\eta)$.

$$p_3 = \frac{2x_1x_2E_p}{x_1 + x_2 - (x_1 - x_2)\cdot\cos(\theta)} \cdot \begin{pmatrix} 1 \\ 0 \\ \sin(\theta) \\ \cos(\theta) \end{pmatrix} \tag{4.6}$$

$$p_4 = \frac{E_p}{x_1 + x_2 - (x_1 - x_2)\cdot\cos(\theta)} \cdot \begin{pmatrix} (x_1^2 + x_2^2) - (x_1^2 - x_2^2)\cos(\theta) \\ 0 \\ -2x_1x_2\sin(\theta) \\ (x_1^2 - x_2^2) - (x_1^2 + x_2^2)\cos(\theta) \end{pmatrix} \tag{4.7}$$

4.3 Implementation and Results

The considerations of Sections 4.1 and 4.2 can now be applied to obtain a cross section and histograms of observables for the scattering of two protons into two photons. Because the PDF is not available in closed form, event generation is the only viable way to obtain distributions of observables and to verify theory against experiment, even with this simple leading-order process.

The integrand in (4.2) can be concertized into (4.8), where q runs over all quarks (except the top quark). The sum has been symmetrized, otherwise a double sum with q and $\bar{q}$ would have been necessary. The choice of Q^2 is justified in Section 4.1 and formulated in (4.9).

$$\frac{\mathrm{d}^3\sigma}{\mathrm{d}\eta\,\mathrm{d}x_1\,\mathrm{d}x_2} = \sum_q \left[f_q(x_1;Q^2)\, f_{\bar{q}}(x_2;Q^2) + f_q(x_2;Q^2)\, f_{\bar{q}}(x_1;Q^2) \right] \frac{\mathrm{d}\sigma(x_1, x_2, Z_q)}{\mathrm{d}\eta} \tag{4.8}$$

$$Q^2 = 2x_1x_2E_p^2 \tag{4.9}$$

The PDF set that is being used in the following has been fitted (and *DGLAP* developed) at leading order and is the central member of the PDF set `NNPDF31_lo_as_0118` provided by *NNPDF* collaboration and accessed through the `LHAPDF6` library [10, 11].

4.3.1 Cross Section

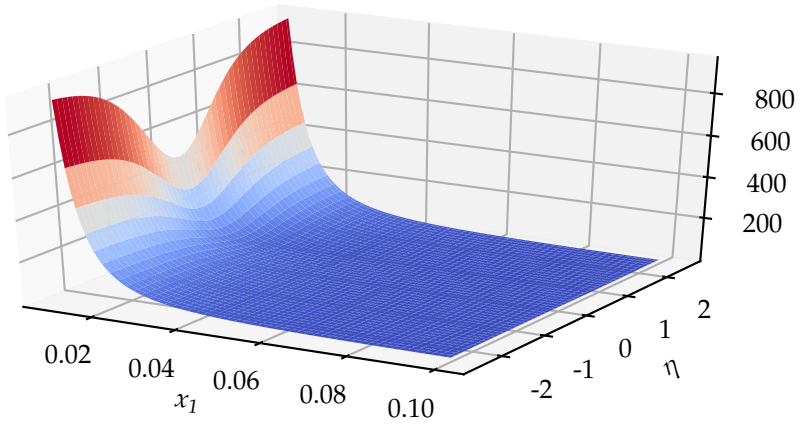
The distribution (4.8) can now be integrated to obtain a total cross-section as described in Section 3.1. For the numeric analysis a proton beam energy of $E_p = 6500\,\mathrm{GeV}$ has been chosen, in resemblance to *LHC* beam energies. The cuts $|\eta| \leq 2.5$ and $p_\mathrm{T} \geq 20\,\mathrm{GeV}$ have been imposed for each photon. Integrating (4.8) with respect to those cuts using `VEGAS` yields $\sigma = (38.685 \pm 0.011)\,\mathrm{pb}$ which is compatible with $\sigma_s = (38.673 \pm 0.037)\,\mathrm{pb}$, the value `Sherpa` gives.

4.3.2 Event Generation and Histograms

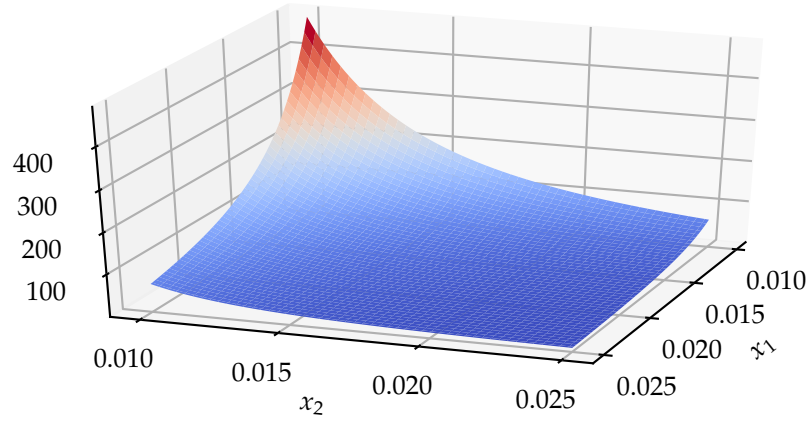
Generating events of $pp \to \gamma\gamma$ is very similar in principle to sampling the partonic cross section. As before, the range of the η parameter has to be constrained to obtain physical

results. Because the absolute values of the pseudo rapidities of the two final state photons are not equal in the lab frame, the shape of the integration/sampling volume differs from a simple hypercube. Furthermore, for the massless limit to be applicable, the center of mass energy of the partonic system must be much greater than the quark masses. This can be implemented by demanding the transverse momentum p_T of the final state photons to be greater than $p_T \geq 20\,\text{GeV}$. A restriction on p_T is suitable, because detectors are usually only sensitive above a certain p_T threshold and the final state particles have to be isolated from the beams.

The resulting distribution (without cuts) is depicted in Fig. 4.1a for fixed x_2 and in Fig. 4.1b for fixed η. For $x_1 = x_2$ the distribution retains some likeness with the partonic distribution (see Fig. 3.1b) but gets suppressed for greater values of x_1. The overall shape of the distribution is clearly highly sub-optimal for hit-or-miss sampling, being very steep and only having significant magnitude when x_1 or x_2 are small (Fig. 4.1b). To remedy that, one has to use a



a) Differential cross section convolved with PDFs for fixed $x_2 = 0.01$ in picobarn.



b) Differential cross section convolved with PDFs for fixed $\eta = 0$ in picobarn.

Figure 4.1: Differential cross section convolved with PDFs with one parameter fixed.

more efficient sampling algorithm (`VEGAS`) or impose very restrictive cuts. The self-coded implementation used here can be found as described in Section 1.2 and employs `VEGAS`-optimized stratified sampling (as discussed in Section 3.2.4) and the hit-or-miss method. The

matrix element (ME) and cuts are implemented using `Cython` [12] to obtain better performance, as these terms are evaluated very often. The ME and the cuts are then convolved with the PDF (as in (4.8)) and wrapped into a simple function with a generic interface and plugged into the `VEGAS` implementation which then computes the integral, grid, the individual weights of the grid hypercubes and rough estimates of the maxima in each hypercube. In principle the code could be generalized to other processes by simply redefining the matrix elements, as no other part of the code is process specific. The cuts work as simple θ-functions, which has the advantage, that the maximum for hit or miss can be chosen with respect to those cuts. On the other hand, this method introduces discontinuity into the integrand, which is problematic for numeric maximizers. The estimates of the maxima, provided by the `VEGAS` implementation, are used as the starting point for a gradient ascend maximizer. In this way, the discontinuities introduced by the cuts are being circumvented. Because the stratified sampling requires very accurate upper bounds, they have been overestimated by 1 %, which lowers the efficiency slightly but reduces bias. The sampling algorithm chooses hypercubes randomly in accordance to their contribution to the integral by generating a uniformly distributed random number $r \in [0, 1]$ and summing the weights of the hypercubes until the sum exceeds this number. The last hypercube in this sum is then chosen and one sample is obtained. Taking more than one sample can improve performance, but introduces bias, as hypercubes with low weight may be oversampled. At various points, the `numba` [13] package has been used to just-in-time compile code to increase performance. The `Python` `multiprocessing` module is used to parallelize the sampling and exploit all CPU cores. Although the `VEGAS` step is very (*very*) time intensive, the actual sampling performance is in the same order of magnitude as `Sherpa`.

A sample of $N = 10000000$ events has been generated both in `Sherpa` (with the same cuts) and through own code. The resulting histograms of some observables are depicted in Fig. 4.2. The sampling efficiency achieved was $\mathfrak{e} = 33\,\%$ using a total of $K = 57600$ hypercubes.

The histograms are more or less compatible with each other [4]. In all cases the difference between T-Value and the mean of the χ^2 distribution for that value (= 50, the number of bins) is less then the standard deviation (= 10) of the same distribution and thus the histograms are considered compatible. The angular distributions for $\eta, \cos\theta$ show agreeable P-values, but the very steep distributions for p_T and $m_{\gamma\gamma}$ are especially sensitive to fluctuations and the systemic errors introduced of the weight of each hypercube. Therefore their formal measure of compatibility, the P-Value, is rather low. This indicates that the MC error in the determination of the weights for the hypercubes should be studied more carefully and highlights the disadvantage of the sampling method chosen here. The kinematics and PDF values were compared with `Sherpa` and proved to be equivalent.

The `Sherpa` runcard utilized here and the analysis used to produce the histograms can be found in Appendices A.1.2 and A.2.2. When comparing Figs. 3.6a and 4.2a it becomes apparent, that the PDF has substantial influence on the resulting distribution. Also the center of momentum energy is not constant anymore and has a steep peak at low energies due to the steepness of the PDF. The convolution with the PDF has also smoothed out the Jacobian peak seen in Fig. 3.6b.

Furthermore new observables have been introduced. The invariant mass of the photon pair $m_{\gamma\gamma} = (p_{\gamma,1} + p_{\gamma,2})^2$ is the center of mass energy of the partonic system that produces the photons (see (4.3)) and proportional to the product of the momentum fractions of the partons.

[4] See Appendix A.3.2 for a description of the compatibility test.

Figure 4.2d shows, that the vast majority of the reactions take place at a rather low c.m. energy, owing to the high weights of the PDF at small x values. Due to the p_T cuts the first bin is slightly lower than the second.

The cosines of the scattering angles in the lab frame and the Collins-Soper (CS) frame are defined in (4.10) and (4.11).

(4.10)

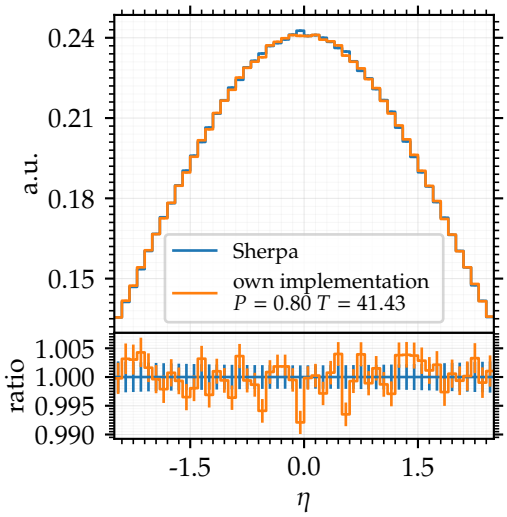
$$\cos\theta^* = \tanh\frac{\eta_1 - \eta_2}{2}$$

(4.11)

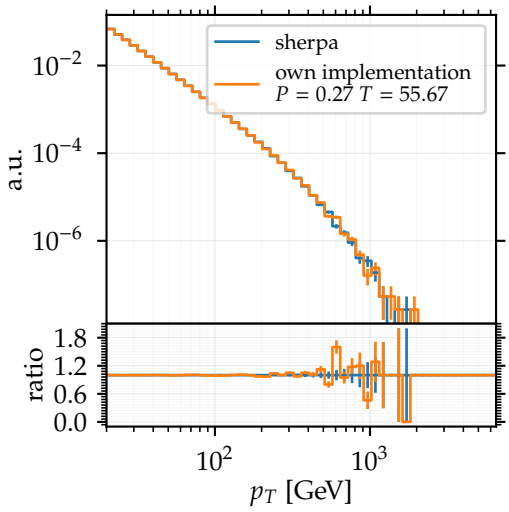
$$\cos\theta^*_{CS} = \frac{\sinh(\eta_1 - \eta_2)}{\sqrt{1 + (p_{T,1} + p_{T,2})^2/m^2_{\gamma\gamma}}} \cdot \frac{2p_{T,1}p_{T,2}}{m^2_{\gamma\gamma}}$$

The scattering angle is just the angle between one photon and the z-axis (beam axis) in the c.m. frame if this frame can be reached by a boost along the z-axis[5]. Here, the partons are assumed to have no transverse momentum and the system is symmetric around the beam axis and therefore this boost is possible. When allowing transverse parton momenta, as will be done in Chapter 5 this symmetry goes away. Defining the z-axis as one beam axis in the c.m. frame would be quite an arbitrary choice that disrespects the symmetry of the two beams considered here (same energy, identical protons). Also the random direction of the transverse momentum can add noise that does not contain much information. The CS frame is defined as the rest frame of the two outgoing photons in which the z-axis bisects the angle between the first beam momentum and the inverse momentum of the second beam. In this frame, which was originally conceived to simplify the extension of the Drell-Yan parton model to transverse parton momenta [14], some symmetry is restored and the study of effects of transverse parton momenta is facilitated. Because of the above-mentioned symmetry, the histograms in Figs. 4.2e and 4.2f are the same. One would naively expect some likeness to Fig. 3.3 but the cuts imposed alter the distribution quite considerably, cutting off the $\cos\theta^* \to 1$ region.

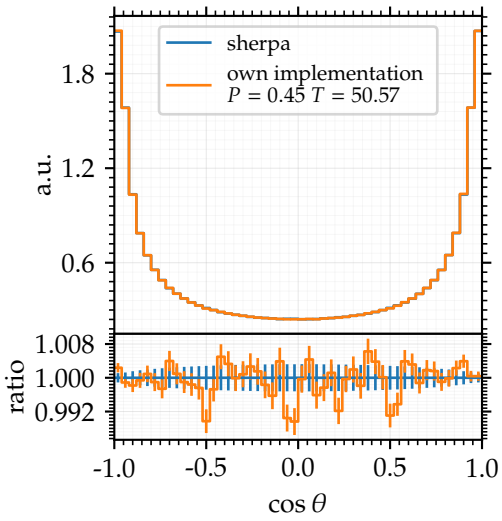
[5]Or more generally, in a z-boosted frame where the polar angles of the two photons are the same.



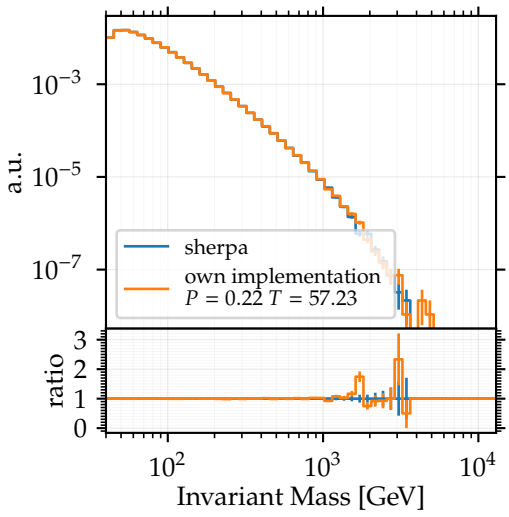
a) η distribution.



b) p_T distribution.

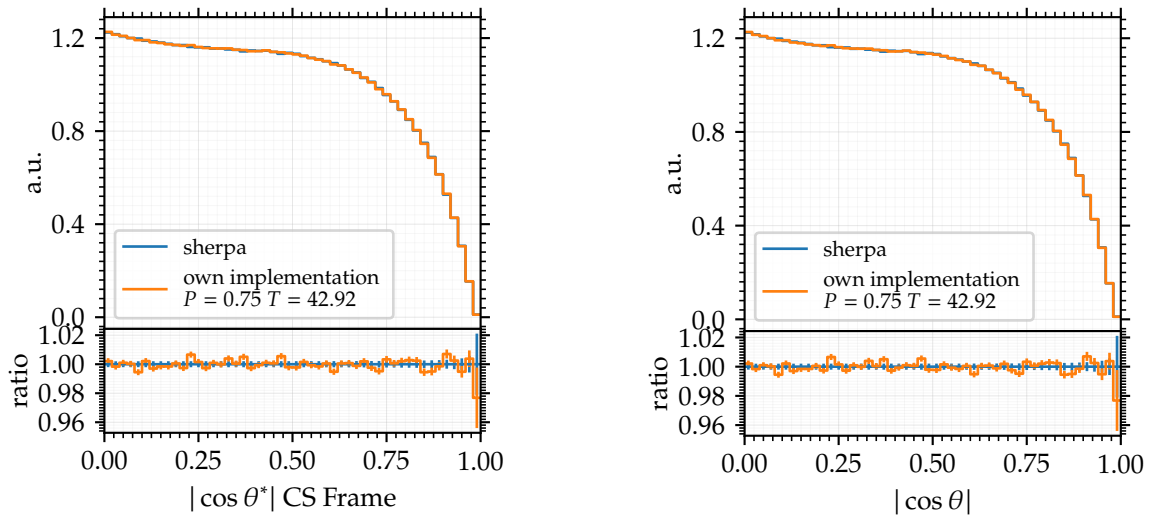


c) $\cos\theta$ distribution.



d) Invariant mass of the final state photon system.

Figure 4.2: Continued on next page.



e) Scattering angle of the two photons in the CS frame. f) Scattering angle of the two photons in the lab frame.

Figure 4.2: Comparison of histograms of observables for $pp \to \gamma\gamma$ generated manually and by `Sherpa/Rivet` and normalized to unity. The sample size was $N = 10000000$.

5 Phenomenological Studies of the Diphoton Process in Proton-Proton Scattering

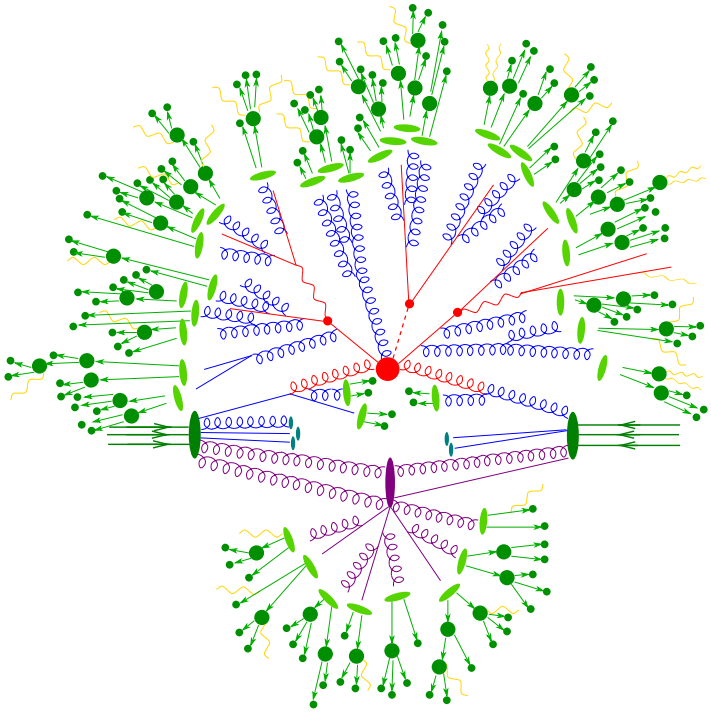


Figure 5.1: The anatomy of a $t\bar{t}h$ event. Taken from [4]. Described below.

In real proton scattering the hard process discussed in Chapter 4 is only a part of the whole event. Figure 5.1 shows a pictorial representation of the plethora of processes taking place around the hard event (red), marking distinct processes in the event generator by different colors. Partons do in general have some intrinsic transverse momentum. Scattered charges radiate in both QCD and QED, both giving rise to shower-like cascades (blue, yellow) and both can lead to additional transverse momentum of the initial state partons. The remnants of the proton can radiate showers themselves, scatter in more or less hard processes (Multiple Interactions, MI, purple blob) and affect the hard process through color correlation. Finally the partons from the showers recombine into hadrons (hadronization, light green blobs) due to QCD confinement. This last effect doesn't produce diphoton-relevant background directly, but affects photon isolation. Those hadrons then decay if they are unstable (dark green blobs), producing even more final state particles. All of those processes have to be taken into account to generate events that can be compared with experimental data. [4, 8]

These effects can be calculated or modeled on an per-event base by modern Monte Carlo event generators like `Sherpa`. But these calculations and models are approximations in most cases. This is done for the diphoton process in a gradual way described in Section 5.1. Histograms of observables are generated and discussed in Section 5.2.

5.1 Set-Up and Analysis

To observe the impact on the individual aspect of the proton scattering the following run configurations have been defined. They are incremental in the sense that each subsequent configuration extents the previous one and thus called stages from now on and are listed below.

LO The hard process on parton level as used in Section 4.3.

LO+PS The shower generator of `Sherpa`, *CSShower* [15] (dipole-shower), is activated and simulates initial state radiation. The recoil scheme proposed in [16], which has been proven more accurate for diphoton production at leading order, has been enabled.

LO+PS+pT The beam remnants are simulated, giving rise to additional radiation and parton showers. Also the partons are being assigned primordial p_T, distributed like a Gaussian with a mean value of 0.8 GeV and a standard deviation of 0.8 GeV[1].

LO+PS+pT+Hadronization A cluster hadronization model implemented in *Ahadic* [17] is activated. The shower particles are being hadronized and the decay of the resulting hadrons simulated if they are unstable.

LO+PS+pT+Hadronization+MI Multiple Interactions (MI) based on the Sjöstrand-van-Zijl Model are simulated with *Amisic* [18]. The MI are parton shower corrected, so that there are generally more particles in the final state.

A detailed description of the implementation of those models can be found in [4] and [18].

The runcard implementing the `LO+PS+pT+Hadronization+MI` configuration can be found in Appendix A.1.3. The $|\eta| \leq 2.5$ and $p_T > 20\,\text{GeV}$ cuts are the same as in Section 4.3.1 and the beam energies have been chosen as 6500 GeV to resemble *LHC* conditions. The cuts on the hard process have been loosened to $p_T \geq 5\,\text{GeV}$ and $|\eta| \leq 3$ because jets and primordial p_T can increase the final state p_T to fall into the analysis cuts.

The analysis is similar to the one used in Section 4.3.2 with the addition of some observables arising from the possibility of non-zero transverse momentum of the photon pair:

- total transverse momentum of the photon pair
- azimuthal angle between the two photons
- transverse momentum of the sub-leading photon (see below)

[1] Those values are `Sherpa`'s defaults.

Because the final state now potentially contains additional photons from hadron decays, the analysis only selects prompt photons with the highest p_T (leading photons). Furthermore a cone of

$$R = \sqrt{(\Delta\varphi)^2 + (\Delta\eta)^2} \leq 0.4$$

around each photon must not contain more than 4.5 % of the photon transverse momentum (+6 GeV), to simulate experimental photon isolation. The leading photons are required to have $\Delta R > 0.45$, to filter photons with overlapping isolation cones, which would be hard to isolate in experiments. In truth, the analysis already excludes such photons, but to be compatible with experimental data, which must rely on such criteria, they have been included. These cuts are called *isolation cuts*. The code of the analysis is listed in Appendix A.2.3 and has been adapted from code privately communicated by Frank Siegert and Heberth Torres.

The production of photons in showers has not been considered.

5.2 Discussion

		events discarded by cuts	
Stage	σ [pb]	phase space [%]	isolation [1×10^{-4} %]
`LO+PS+pT+Hadr.+MI`	33.02 ± 0.07	97.63	9.56
`LO+PS+pT+Hadr.`	34.08 ± 0.07	97.56	1.89
`LO+PS+pT`	33.97 ± 0.07	97.56	3.52
`LO+PS`	34.60 ± 0.07	97.52	3.63
`LO`	38.74 ± 0.07	96.77	0

Table 5.1: Cross sections and cut statistics.

The results of the `Sherpa` runs for each stage with 10^7 events each are depicted in the histograms in Fig. 5.2 and shall now be discussed in detail.[2] Because of the analysis cuts, the total number of accepted events is smaller than the number of events generated by `Sherpa`, but sufficient for proper statistics for most observables. Also the fiducial cross sections of the different stages, which are compared in Table 5.1, differ as a result of the NLO effects that have been switched on. All histograms are normalized to their respective cross sections.

When there is no additional p_T, then the photon momenta are back to back in the plane perpendicular to the beam axis (transverse plane). Because four-momentum conservation is enforced, every emission from a parton gives a recoil momentum to that parton. If the system now gets a recoil from parton showering, then this usually subtracts p_T from one of the photons unless that recoil is near perpendicular to the photons. This leads to an increase in rejected events. Because the p_T distribution of the photons (Figs. 5.2c and 5.2d) is very steep, a lot of events produce photons with low p_T that a prone to falling under the cuts and so this effect is substantial. The fraction of events that have been discarded by the phase space cuts

[2] An even higher precision study was not possible due to unavailability of access to the *Taurus* cluster at the time of writing.

are listed in Table 5.1 which shows an increase in the fraction of discarded events for all stages after LO, contributing to the drop in fiducial cross section for the LO+PS and LO+PS+pTstages.

The isolation cuts do affect the observed cross section as well, as is shown in Table 5.1. The fiducial LO+PS+pT+Hadr. cross section is a bit higher than the LO+PS+pT one, because the hadronization favors isolation of photons by reducing the collinearity of the particles in the final state and may create particles like neutrinos which are not detected in the calorimeter or can be easily identified (muons). The opposite effect can be seen with MI, where the number of final state particles and thus the hadronic activity in the isolation cone is increased. This effect leads to another drop in the fiducial cross section.

The transverse momentum of the photon system (see Fig. 5.2a) now becomes non trivial, as both the LO+PS and LO+PS+pT stages affect this observable directly. Initial state radiation generated by the parton showering algorithm kicks the quarks involved in the hard process and thus generates transverse momentum. In regions of high p_T all but the LO stage are largely compatible, falling off steeply at $\mathcal{O}(10\,\text{GeV})$. Because the parton shower approximation is only valid in the collinear limit, it can't necessarily be trusted in higher p_T regions [8]. The fact that the distribution has a maximum and falls again towards lower p_T is related to the nature of parton shower algorithms, which approximately sum over all terms of a perturbation series [8].

The partons in a proton are somewhat localized and thus the uncertainty principle demands that those partons have some momentum perpendicular to the proton motion. The default parameters in Sherpa assign transverse momenta according to a Gaussian distribution with a mean and standard deviation of 0.8 GeV. In the region of 1 GeV and below, the effects of primordial p_T show as an enhancement in the cross section of the LO+PS+pT stage.

Related effects can be seen in the distribution for the azimuthal separation of the photons in Fig. 5.2b. Back to back photons are favored by all stages because deviations from this configuration are purely higher order effects, so most events feature an azimuthal separation of less than $\pi/2$. The enhancement of the low p_T regions in the LO+PS+pT stage also leads to an enhancement in the back-to-back region for this stage over the LO+PS stage.

In the p_T distribution of the leading photon (see Fig. 5.2c) the boost of the leading photon towards higher p_T in all stages but the LO stage originates from the parton showering and thus the distribution of those stages are largely compatible beyond 1 GeV. Again, the effect of primordial p_T becomes visible for transverse momenta smaller than 1 GeV.

The p_T distribution for the sub-leading photon (see Fig. 5.2d) shows remarkable resemblance to the LO distribution for all other stages, although there is a very minute bias to lower p_T. This is consistent with the mechanism described above so that events that subtract (very small amounts of) p_T from the sub-leading second photon are more common. Interestingly, the effects of primordial p_T are not very visible.

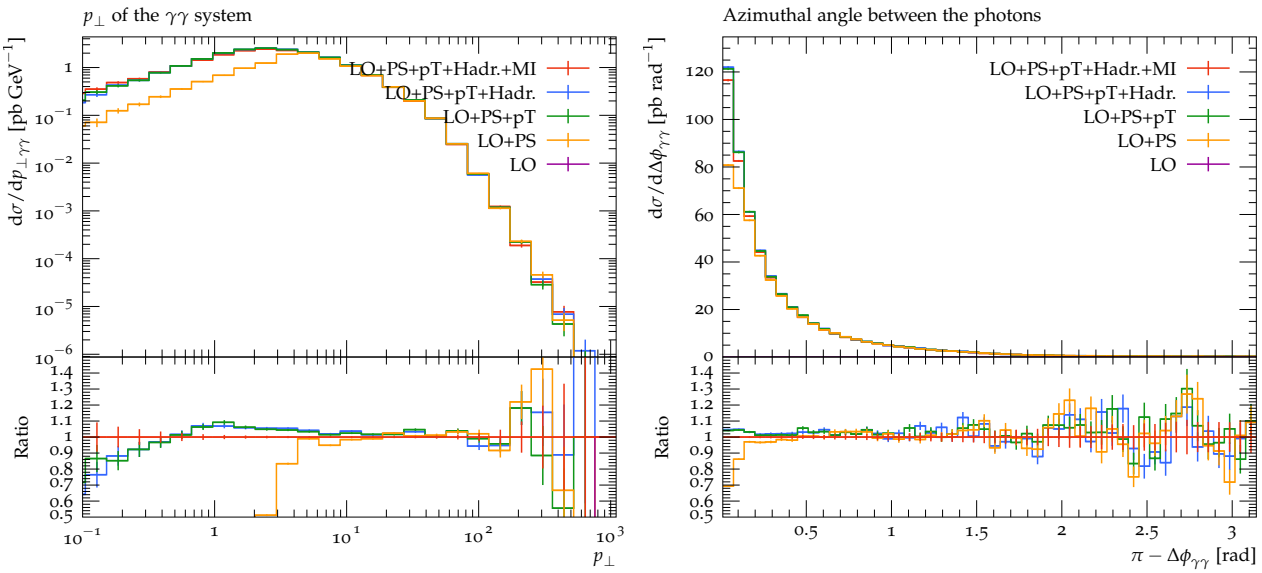
In leading order, the phase space cuts impose a hard lower bound to the invariant mass of the photon system. Higher order effects can give recoil momentum to the partons in such a way, that events with lower invariant mass pass the cuts. The distribution for the invariant mass (see Fig. 5.2e) shows that effect. The decline of the cross section towards lower energies is much steeper than the PDF-induced decline towards higher energies. High p_T boost in a are very rare, which supports the reasoning about the drop in fiducial cross section. Also, due to

the implementation of the showering algorithm, it may be that only the rather small intrinsic p_T changes the center of momentum energy at all.

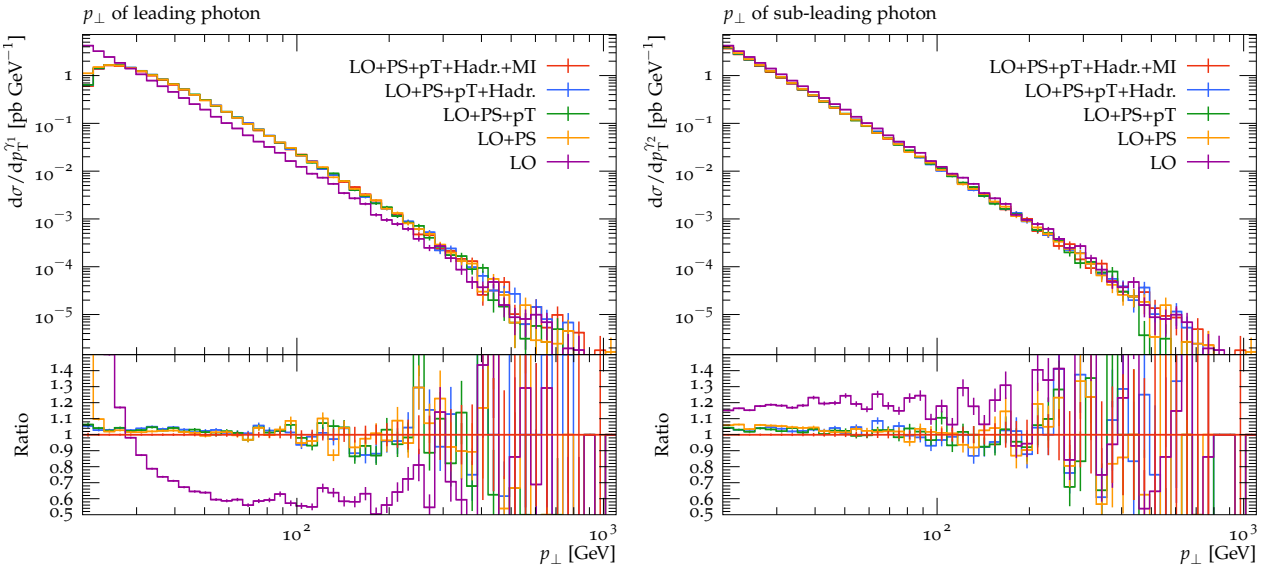
The angular distributions of the leading photon in Figs. 5.2f and 5.2g are most affected by the differences in total cross section and slightly shifted towards more central (transverse) regions for all stages from LO+PS on due to the p_T kicks to the photon system.

Because the diphoton system itself is only affected by recoil to the initial state quarks, the scattering angle cross sections in Figs. 5.2h and 5.2i show a similar shape in all stages. Towards small scattering angles, the differences in shape grow larger, as this is the region where the cuts have the largest effect. In the CS frame, the cross section does not converge to zero for LO+PS and subsequent stages. With non-zero p_T of the photon system, the z-axis of the CS frame rotates out of the region that is affected by cuts. The ratio plot also shows, that the region where cross section distributions are similar in shape extends further. In the CS frame effects of the non-zero p_T of the photon system are (somewhat weakly) suppressed.

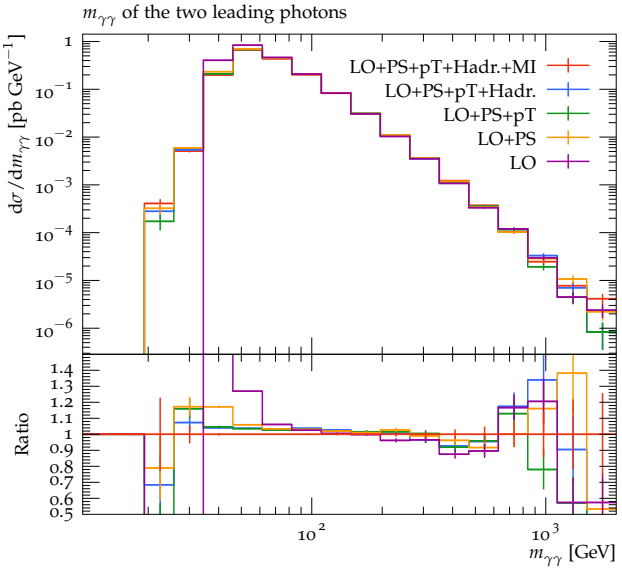
It becomes clear, that parton showering and the primordial transverse momentum have the biggest effect on the shape of observables, as they affect the kinematics of the diphoton process directly. Isolation effects show most through hadronization and especially multiple interactions. In observables that exist in leading order (η, p_T, ...), the hard process alone gives a reasonably good qualitative picture, but in most other observables higher order effects introduce considerable deviations and have to be taken into account for a realistic study, even with this simple process.


$p_\perp$ of the $\gamma\gamma$ system
$d\sigma/dp_{\perp\gamma\gamma}$ [pb GeV^{-1}]
LO+PS+pT+Hadr.+MI
LO+PS+pT+Hadr.
LO+PS+pT
LO+PS
LO
Ratio
$p_\perp$
Azimuthal angle between the photons
$d\sigma/d\Delta\phi_{\gamma\gamma}$ [pb rad^{-1}]
$\pi - \Delta\phi_{\gamma\gamma}$ [rad]

a) b)


$p_\perp$ of leading photon
$d\sigma/dp_T^{\gamma 1}$ [pb GeV^{-1}]
LO+PS+pT+Hadr.+MI
LO+PS+pT+Hadr.
LO+PS+pT
LO+PS
LO
Ratio
$p_\perp$ [GeV]
$p_\perp$ of sub-leading photon
$d\sigma/dp_T^{\gamma 2}$ [pb GeV^{-1}]

c) d)


$m_{\gamma\gamma}$ of the two leading photons
$d\sigma/dm_{\gamma\gamma}$ [pb GeV^{-1}]
LO+PS+pT+Hadr.+MI
LO+PS+pT+Hadr.
LO+PS+pT
LO+PS
LO
Ratio
$m_{\gamma\gamma}$ [GeV]

e)

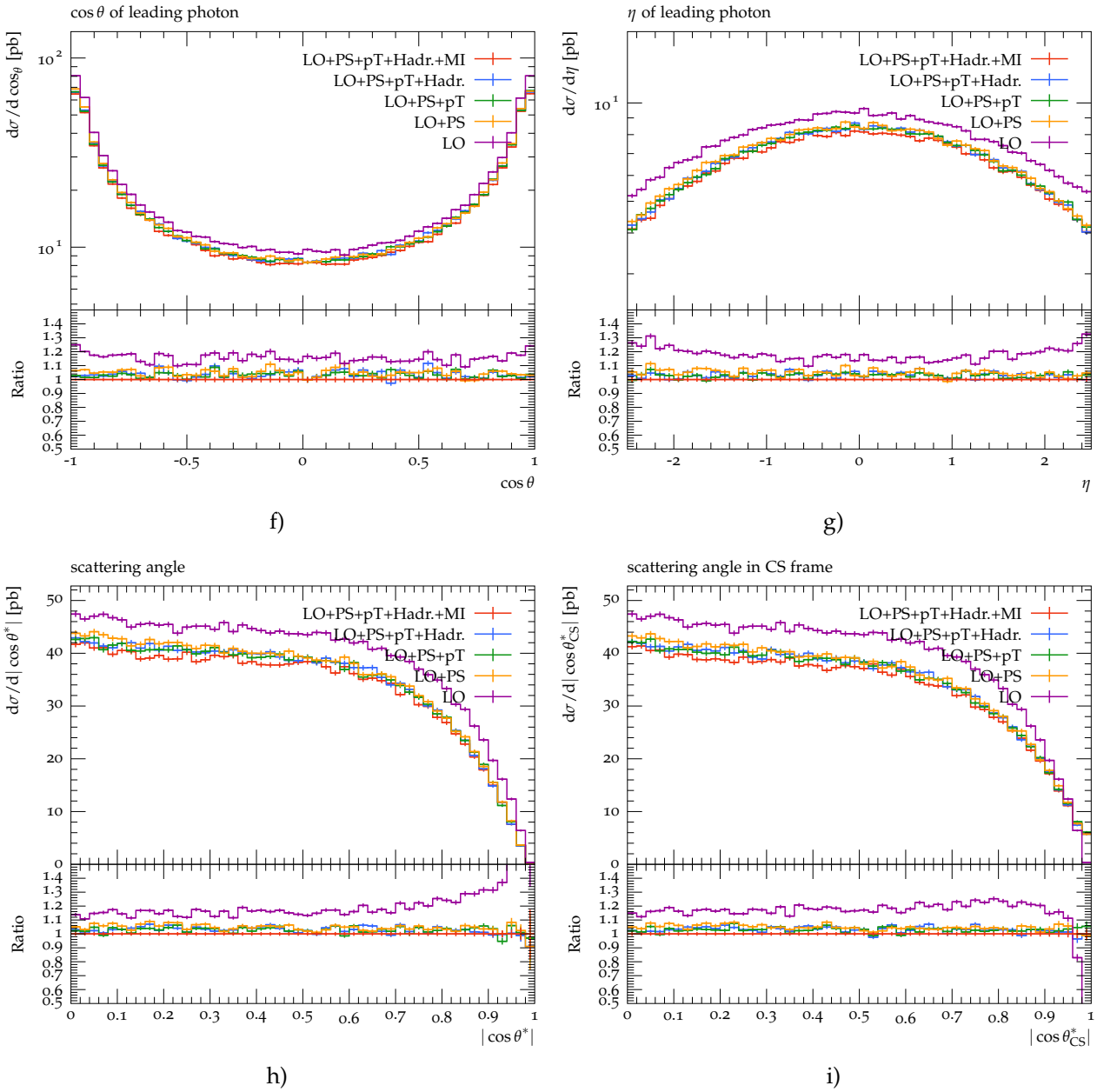


Figure 5.2: Histograms of observables generated by simulations with increasingly more effects turned on.

6 Summary and Outlook

In this thesis, the leading order analytical cross section for the quark level diphoton process has been calculated and verified using the `Sherpa` event generator. Subsequently some Monte Carlo methods for integration and sampling were mathematically motivated, implemented and applied to the diphoton process, resulting in the implementation of a simple event generator for proton-proton scattering. Good sampling efficiency was achieved and potential problems with the employed algorithm were highlighted.

Finally a phenomenological study of the diphoton process in proton-proton scattering was performed by incrementally enabling additional effects in the `Sherpa` event generator. Albeit the leading order matrix element gives a good qualitative picture for the shape of some observables, higher order effects like parton showering proved to have a significant impact on certain observables.

The simplistic implementation of the diphoton process could be developed further by using NLO matrix elements for the hard process. This would lead to extra emissions and new requirements for photon isolation and a plethora of new effects. Another avenue of refinement of the simulation would be to allow the creation of photons in parton showers. The impact of increased photon activity could lead to additional observable effects.

A Appendix

A.1 Sherpa Runcards

A.1.1 Quark Antiquark Anihilation

```
# general options
OUTPUT: 3

# event count
EVENTS: 1000000

# save results
GENERATE_RESULT_DIRECTORY: true

# set up d d̄ beams, each at 100 GeV
BEAMS: [1, -1]
BEAM_ENERGIES: 100

# matrix-element calculation
ME_GENERATORS:
- Comix
- Amegic

# massless d
PARTICLE_DATA:
  1:
    Massive: false

# the main scattering process
PROCESSES:
- 1 -1 -> 22 22:
    Order: {QCD: 0, EW: 2}
    Integration_Error: 0.001

# use pure QED
EW_SCHEME: alpha0

# disable all advanced processes
FRAGMENTATION: None
SCALES: 'VAR{2*Abs2(p[0]+p[1])}'
SHOWER_GENERATOR: None
MPI: None
```

```
38
39 # cut to |η| ≤ 2.5
40 SELECTORS:
41 - [Eta, 22, -2.5, 2.5]
42
43 # no transverse impulses
44 BEAM_REMNANTS: false
45
46 ANALYSIS: Rivet
47 RIVET:
48   ANALYSES:
49     - MC_DIPHOTON_SIMPLE
```

A.1.2 Proton Proton Scattering

```
1  # general options
2  OUTPUT: 3
3
4  # event count
5  EVENTS: 10000000/8  # / num_cpus
6
7  RESULT_DIRECTORY: out/Results
8  ANALYSIS_OUTPU: out/Analysis
9
10 # save results
11 GENERATE_RESULT_DIRECTORY: true
12
13 # set up p p̄ beams, each at 6500 GeV
14 BEAMS: [2212, 2212]
15 BEAM_ENERGIES: 6500
16 PDF_LIBRARY: LHAPDFSherpa
17 PDF_SET: NNPDF31_lo_as_0118
18 PDF_SET_VERSIONS: [0, 0]
19
20 # matrix-element calculation
21 ME_GENERATORS:
22 - Comix
23 - Amegic
24
25 SCALES: 'VAR{1/2*Abs2(p[0]+p[1])}'
26
27 # the main scattering process
28 PROCESSES:
29 - 94 -94 -> 22 22:
30     Order: {QCD: 0, EW: 2}
31     Integration_Error: 0.001
32
33 # use pure QED
34 EW_SCHEME: alpha0
```

```
35
36 # disable all advanced processes
37 FRAGMENTATION: None
38 SHOWER_GENERATOR: None
39 MI_HANDLER: None
40
41 # cuts
42 SELECTORS:
43 - [Eta, 22, -2.5, 2.5]
44 - [PT, 22, 20, 200000000]
45
46 # no transverse impulses
47 BEAM_REMNANTS: false
48
49 ANALYSIS: Rivet
50 RIVET:
51   IGNOREBEAMS: 1
52   USE_HEPMC_NAMED_WEIGHTS: 0
53   ANALYSES:
54     - MC_DIPHOTON_PROTON
```

A.1.3 Holistic Proton Proton Scattering

```
(run){
  % me generator settings
  ME_SIGNAL_GENERATOR Comix;
  EVENT_GENERATION_MODE Unweighted;

  % collider setup
  BEAM_1 2212; BEAM_ENERGY_1 = 6500.;
  BEAM_2 2212; BEAM_ENERGY_2 = 6500.;

  PDF_LIBRARY LHAPDFSherpa;
  PDF_SET NNPDF31_lo_as_0118;
  PDF_SET_VERSION 0;
  ALPHAQED_DEFAULT_SCALE 0;

% TEMPLATE
EW_SCHEME 0;
FRAGMENTATION Ahadic;
SHOWER_GENERATOR CSS;
CSS_KIN_SCHEME=0
MI_HANDLER Amisic;
BEAM_REMNANTS 1;
}(run)

(processes){
  Process 94 -94 -> 22 22;
```

```
  Order (*,2);
  Integration_Error 0.001 {2};
  End process;
}(processes)

(selector){
  PseudoRapidity 22 -3 3;
  PT 22 5 200000000;
}(selector)

(analysis){
  BEGIN_RIVET {
    -a MC_DIPHOTON_PROTON;
    IGNOREBEAMS 1;
    USE_HEPMC_NAMED_WEIGHTS 0;
  } END_RIVET
}(analysis)
```

A.2 Rivet Analysis Code

A.2.1 Simple Diphoton Analysis

```
// -*- C++ -*-
#include "Rivet/Analysis.hh"
#include "Rivet/Projections/DressedLeptons.hh"
#include "Rivet/Projections/FastJets.hh"
#include "Rivet/Projections/FinalState.hh"
#include "Rivet/Projections/MissingMomentum.hh"
#include "Rivet/Projections/PromptFinalState.hh"

namespace Rivet {

/// @brief Generate some simple histograms of the diphoton process. TODO: more
class MC_DIPHOTON_SIMPLE : public Analysis {
public:
  /// Constructor
  DEFAULT_RIVET_ANALYSIS_CTOR(MC_DIPHOTON_SIMPLE);

  /// @name Analysis methods
  //@{

  /// Book histograms and initialise projections before the run
  void init() {
    // Initialise and register projections

    // The basic final-state projection:
    // all final-state particles within
```

```
26      // the given eta acceptance
27      // We hardcode this TODO: make this configurable
28      FinalState fs;
29      declare(fs, "FS");
30
31      // cut has been made in sherpa
32      IdentifiedFinalState ifs {};
33      ifs.acceptId(PID::PHOTON);
34      declare(ifs, "IFS");
35
36      auto energy = info().energies()[0].first;
37      book(_h_pT, "pT", 50, 16.3, energy);
38      book(_h_eta, "eta", 50, -2.5, 2.5);
39      book(_h_cos_theta, "cos_theta", 50, -1, 1);
40    }
41
42    /// Perform the per-event analysis
43    void analyze(const Event &event) {
44      const double weight = 1.0;
45
46      Particles photons = apply<IdentifiedFinalState>(event, "IFS").particles();
47
48      // they are both the same, so we take the first
49      const auto &photon = photons.front();
50
51      _h_pT->fill(photon.pT(), weight);
52      _h_eta->fill(photon.eta(), weight);
53      _h_cos_theta->fill(cos(photon.theta()), weight);
54    }
55
56    //@}
57
58    void finalize() {
59      normalize(_h_pT);
60      normalize(_h_eta);
61      normalize(_h_cos_theta);
62    }
63
64    /// @name Histograms
65    //@{
66    Histo1DPtr _h_pT;
67    Histo1DPtr _h_eta;
68    Histo1DPtr _h_cos_theta;
69    //@}
70  };
71
72  DECLARE_RIVET_PLUGIN(MC_DIPHOTON_SIMPLE);
73
74  } // namespace Rivet
```

A.2.2 Proton Proton Scattering Analysis

```
// -*- C++ -*-
#include "Rivet/Analysis.hh"
#include "Rivet/Projections/DressedLeptons.hh"
#include "Rivet/Projections/FastJets.hh"
#include "Rivet/Projections/FinalState.hh"
#include "Rivet/Projections/MissingMomentum.hh"
#include "Rivet/Projections/PromptFinalState.hh"

namespace Rivet {

/// @brief Generate some simple histograms of the diphoton process. TODO: more
class MC_DIPHOTON_PROTON : public Analysis {
public:
  /// Constructor
  DEFAULT_RIVET_ANALYSIS_CTOR(MC_DIPHOTON_PROTON);

  /// @name Analysis methods
  //@{

  /// Book histograms and initialise projections before the run
  void init() {
    // Initialise and register projections

    // The basic final-state projection:
    // all final-state particles within
    // the given eta acceptance
    // We hardcode this TODO: make this configurable
    FinalState fs;
    declare(fs, "FS");

    // cut has been made in sherpa
    IdentifiedFinalState ifs{};
    ifs.acceptId(PID::PHOTON);
    declare(ifs, "IFS");

    auto energy = info().energies()[0].first;
    double min_pT = 20;
    double eta = 2.5;

    _observables = {"pT", "eta", "cos_theta", "inv_m", "o_angle", "o_angle_cs"};

    book(_histos["pT"], "pT", logspace(50, min_pT, energy, true));
    book(_histos["eta"], "eta", 50, -eta, eta);
    book(_histos["cos_theta"], "cos_theta", 50, -1, 1);
    book(_histos["inv_m"], "inv_m", logspace(50, 2 * min_pT, 2 * energy, true));
    book(_histos["o_angle"], "o_angle", 50, 0, 1);
    book(_histos["o_angle_cs"], "o_angle_cs", 50, 0, 1);
  }
```

```
49
50    /// Perform the per-event analysis
51    void analyze(const Event &event) {
52      const Particles &photons =
53          apply<IdentifiedFinalState>(event, "IFS").particles();
54
55      // make sure that there are only two photons
56      if (photons.size() != 2)
57        vetoEvent;
58
59      // they are both the same, so we take the first
60      const auto &photon = photons.front();
61
62      std::map<string, double> obs;
63      obs["pT"] = photon.pT();
64      obs["eta"] = photon.eta();
65      obs["cos_theta"] = cos(photon.theta());
66
67      const auto &moms = photons.moms();
68      const auto total_momentum =
69          std::accumulate(moms.begin(), moms.end(), FourMomentum(0, 0, 0, 0));
70
71      obs["inv_m"] = total_momentum.mass();
72      obs["o_angle"] = std::abs(tanh((photons[1].eta() - photons[0].eta()) / 2));
73
74      //std::abs(photons[0].theta() + photons[1].theta());
75
76      obs["o_angle_cs"] = std::abs(
77          sinh((photons[0].eta() - photons[1].eta())) * 2.0 * photons[0].pT() *
78          photons[1].pT() / sqrt(sqr(obs["inv_m"]) + sqr(total_momentum.pT())) /
79          obs["inv_m"]);
80
81      for (const auto &name : _observables) {
82        _histos[name]->fill(obs[name]);
83      }
84    }
85
86    //@}
87
88    void finalize() {
89      for (auto name : _observables) {
90        normalize(_histos[name]);
91      }
92    }
93
94    /// @name Histograms
95    //@{
96    std::map<string, Histo1DPtr> _histos;
97    std::vector<string> _observables;
98    //@}
99  };
```

```
100
101 DECLARE_RIVET_PLUGIN(MC_DIPHOTON_PROTON);
102
103 } // namespace Rivet
```

A.2.3 Holistic Proton Scattering Analysis

```
1  // -*- C++ -*-
2  #include "Rivet/Analysis.hh"
3  #include "Rivet/Projections/DressedLeptons.hh"
4  #include "Rivet/Projections/FastJets.hh"
5  #include "Rivet/Projections/FinalState.hh"
6  #include "Rivet/Projections/MissingMomentum.hh"
7  #include "Rivet/Projections/PromptFinalState.hh"
8  #include "Rivet/Projections/VetoedFinalState.hh"
9
10 #include <fstream>
11 #include <iostream>
12
13 namespace Rivet {
14
15 struct Observable {
16   double _min, _max;
17   int _bins = 50;
18   bool _log = false;
19   double _value = 0;
20   Histo1DPtr _hist = nullptr;
21
22   // Observable(double min, double max, int bins = 50, bool log = false)
23   //     : _min{min}, _max{max}, _bins{bins}, _log{log} {};
24
25   /// @brief Wraps Histo1D::fill.
26   template <typename... Params> void fill(double value, Params &&... args) {
27     _value = value;
28     if (_hist) {
29       _hist->fill(value, std::forward<Params>(args)...);
30     }
31   }
32 };
33
34 /// @brief Generate some simple histograms of the diphoton process. TODO: more
35 class MC_DIPHOTON_PROTON : public Analysis {
36 public:
37   /// Constructor
38   DEFAULT_RIVET_ANALYSIS_CTOR(MC_DIPHOTON_PROTON);
39
40   /// @name Analysis methods
41   //@{
42
```

```
43    /// Book histograms and initialise projections before the run
44    void init() {
45      // Calorimeter particles for photon isolation
46      VisibleFinalState visFS;
47      VetoedFinalState calo_fs(visFS);
48      calo_fs.addVetoPairId(PID::MUON); // we also want photon
49      declare(calo_fs, "calo_fs");
50
51      // we chain in a prompt final state just to be save
52      PromptFinalState prompt(Cuts::abseta <= 2.5 && Cuts::pT >= 20);
53      IdentifiedFinalState ifs(prompt);
54      ifs.acceptId(PID::PHOTON);
55      declare(ifs, "Photons");
56
57      double min_pT = 20;
58      double eta = 2.5;
59
60      _observables = {{"pT", {min_pT, 1100, 50, true}},
61                      {"pT_subl", {min_pT, 1100, 50, true}},
62                      {"eta", {-eta, eta}},
63                      {"cos_theta", {-1, 1}},
64                      {"inv_m", {0.001, 2000, 50, true}},
65                      {"o_angle", {0, 1}},
66                      {"o_angle_cs", {0, 1}},
67                      {"total_pT", {1e-5, 1100, 50, true}},
68                      {"azimuthal_angle", {1e-2, PI}}};
69
70      for (auto &[name, observable] : _observables) {
71        if (observable._log) {
72          book(observable._hist, name,
73               logspace(observable._bins, observable._min, observable._max));
74          continue;
75        }
76
77        book(observable._hist, name, observable._bins, observable._min,
78             observable._max);
79      }
80
81      book(_xs, "xs");
82      book(_isolation_discard, "isolation_discard");
83      book(_cut_discard, "cut_discard");
84    }
85
86    /// Perform the per-event analysis
87    void analyze(const Event &event) {
88      Particles photons =
89          apply<IdentifiedFinalState>(event, "Photons").particlesByPt();
90
91      // make sure that there are only two photons
92      if (photons.size() < 2)
93        vetoEvent;
```

```
94
95      photons.resize(2);
96
97      // Require the two photons to be separated in dR
98      if (deltaR(photons[0], photons[1]) < 0.45) {
99        _isolation_discard->fill(1);
100       vetoEvent;
101     }
102
103     // require an isolation cone around the photons
104     const Particles fs = apply<VetoedFinalState>(event, "calo_fs").particles();
105     // Loop over photons and require isolation
106     for (const Particle &photon : photons) {
107       // Compute calo isolation via particles within an R=0.4 cone of the photon
108
109       FourMomentum mom_in_EtCone;
110       for (const Particle &p : fs) {
111         // Reject if not in cone
112         if (deltaR(photon.momentum(), p.momentum()) > 0.4)
113           continue;
114         // Sum momentum
115         mom_in_EtCone += p.momentum();
116       }
117
118       // subtract core photon
119       mom_in_EtCone -= photon.momentum();
120
121       // Use photon if energy in isolation cone is low enough
122       if (mom_in_EtCone.Et() > 0.045 * photon.momentum().pT() + 6.0 * GeV) {
123         _isolation_discard->fill(1);
124         vetoEvent;
125       }
126     }
127
128     const auto &photon = photons[0];
129     const auto &photon_subl = photons[1];
130
131     _observables.at("pT").fill(photon.pT());
132     _observables.at("pT_subl").fill(photon_subl.pT());
133     _observables.at("eta").fill(photon.eta());
134     _observables.at("cos_theta").fill(cos(photon.theta()));
135     _observables.at("azimuthal_angle")
136         .fill(PI - mapAngle0ToPi(photon.phi() - photon_subl.phi()));
137
138     const auto &moms = photons.moms();
139     const auto total_momentum = moms[0] + moms[1];
140     _observables.at("total_pT").fill(total_momentum.pT());
141     _observables.at("inv_m").fill(total_momentum.mass());
142
143     _observables.at("o_angle").fill(
144         std::abs(tanh((photons[1].eta() - photons[0].eta()) / 2)));
```

```
145
146       _observables.at("o_angle_cs")
147           .fill(std::abs(sinh((photons[0].eta() - photons[1].eta())) * 2.0 *
148                          photons[0].pT() * photons[1].pT() /
149                          sqrt(sqr(_observables.at("inv_m")._value) +
150                               sqr(total_momentum.pT())) /
151                          _observables.at("inv_m")._value));
152
153       // Fill fiducial cross section
154       _xs->fill();
155     }
156
157     //@}
158
159     void finalize() {
160       const double sf = crossSection() / (picobarn * sumOfWeights());
161       scale(_xs, sf);
162
163       // This is a hack to plot some statistic about discarded events
164       if (_isolation_discard->numEntries() > 0)
165         _isolation_discard->scaleW(_isolation_discard->numEntries() /
166                                    _isolation_discard->val() / numEvents());
167
168       for (int i = 0;
169            i < (numEvents() - _xs->numEntries() - _isolation_discard->val()); i++)
170         _cut_discard->fill(1);
171
172       _cut_discard->scaleW(_cut_discard->numEntries() / _cut_discard->val() /
173                            numEvents());
174
175       for (auto const &[_, observable] : _observables) {
176         scale(observable._hist, sf);
177       }
178     }
179
180     /// @name Histograms
181     //@{
182     std::map<const std::string, Observable> _observables;
183     CounterPtr _xs;
184     CounterPtr _isolation_discard;
185     CounterPtr _cut_discard;
186     //@}
187   };
188
189   DECLARE_RIVET_PLUGIN(MC_DIPHOTON_PROTON);
190
191   } // namespace Rivet
```

A.3 Mathematical Notes

A.3.1 Equivalence of Importance Sampling and Change of Variables

Assume the same prerequisites as in Section 3.1. Here the proof is made for two dimensions, higher dimension follow analogous (but with a burden of notation). In truth, a multidimensional integral can always be reduced to a series of one dimensional integrals, but doing the calculation for the two dimensional case is illustrative.

Define a variable transformation as in (A.1), where the inverses are taken with reference to the variables before the semicolon and $a, b \in [0, 1]$. The inverse can be taken, as $\rho > 0$ is assumed and so all integrals are monotonic.

$$
\begin{aligned}
R_x(x) &= \int_y \int_0^x \rho(x, y)\,\mathrm{d}x\,\mathrm{d}y \\
R_y(y; x) &= \frac{\int_0^y \rho(x, y)\,\mathrm{d}x\,\mathrm{d}y}{\int_y f(x, y)\,\mathrm{d}y} \qquad \text{(A.1)} \\
\mathbf{x} = \begin{pmatrix} x \\ y \end{pmatrix} &= \begin{pmatrix} R_x^{-1}(a) \\ R_y^{-1}(b, x(a)) \end{pmatrix}
\end{aligned}
$$

The Jacobian determinant is thus given by (A.2).

$$
(\partial_a R_x^{-1}(a)) \cdot (\partial_b R_y^{-1}(b; x(a))) = \frac{1}{\int_y \rho(x(a), y)\,\mathrm{d}y} \cdot \frac{\int_y \rho(x(a), y)\,\mathrm{d}y}{\rho(x(a), y(a, b))} = \frac{1}{\rho(x(a), y(a, b))} \qquad \text{(A.2)}
$$

The integral (3.1) becomes (A.3) which is the same as if f (interpreted as a probability density) was transformed to the variables a, b.

$$
\int_\Omega \left[\frac{f(\mathbf{x})}{\rho(\mathbf{x})}\right] \rho(\mathbf{x})\,\mathrm{d}\mathbf{x} = \int_0^1 \int_0^1 \frac{f(x(a), y(a, b))}{\rho(x(a), y(a, b))}\,\mathrm{d}a\,\mathrm{d}b \qquad \text{(A.3)}
$$

That means taking sampling points $\{\mathbf{x_i}\} \sim \rho$ and weighting samples with $1/\rho(\mathbf{x})$ is equivalent to taking uniformly distributed samples of f in a, b space with the appropriate transformation.

This works, because the transformation law (A.1) uniformly distributed samples into samples distributed like ρ.

Given a variable transformation, one can reconstruct a corresponding probability density, by chaining the Jacobian with the inverse of that transformation.

A.3.2 Compatibility of Histograms

The compatibility of histograms is tested as described in [19]. The test value is

$$T = \sum_{i=1}^{k} \frac{(u_i - v_i)^2}{u_i + v_i}$$

where u_i, v_i are the number of samples in the i-th bin of the histograms u, v and k is the number of bins. This value is χ^2 distributed with k degrees, when the number of samples in the histogram is reasonably high. The mean of this distribution is k and its standard deviation is $\sqrt{2k}$. The value

$$P = 1 - \int_0^T f(x; k)\,\mathrm{d}x$$

states with which probability the T value would be greater than the obtained one, where f is the probability density of the χ^2 distribution. Thus $P \in [0, 1]$ is a measure of confidence for the compatibility of the histograms. These formulas hold, if the total number of events in both histograms is the same.

Bibliography

[1] G. Aad et al. (ATLAS), "Observation of a new particle in the search for the Standard Model Higgs boson with the ATLAS detector at the LHC", Phys.Lett. **B716**, 1–29 (2012).

[2] S. Chatrchyan et al. (CMS), "Observation of a new boson at a mass of 125 GeV with the CMS experiment at the LHC", Phys.Lett. **B716**, 30–61 (2012).

[3] M. Aaboud, G. Aad, B. Abbott, et al., "Search for higgs boson pair production in the $\gamma\gamma b\bar{b}$ final state with 13 tev pp collision data collected by the atlas experiment", Journal of High Energy Physics **2018**, 10.1007/jhep11(2018)040 (2018).

[4] T. Gleisberg, S. Hoeche, F. Krauss, M. Schonherr, S. Schumann, F. Siegert, and J. Winter, "Event generation with SHERPA 1.1", JHEP **02**, 007 (2009).

[5] P. Lepage, "A new algorithm for adaptive multidimensional integration", Journal of Computational Physics **27**, 192–203 (1978).

[6] C. Bierlich et al., "Robust Independent Validation of Experiment and Theory: Rivet version 3", SciPost Phys. **8**, 026 (2020).

[7] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. Jarrod Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. Carey, . Polat, Y. Feng, E. W. Moore, J. Vand erPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt, and S. 1. 0. Contributors, "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python", Nature Methods **17**, 261–272 (2020).

[8] A. Buckley, J. Butterworth, S. Gieseke, D. Grellscheid, S. Höche, H. Hoeth, F. Krauss, L. Lönnblad, E. Nurse, P. Richardson, S. Schumann, M. H. Seymour, T. Sjöstrand, P. Skands, and B. Webber, "General-purpose event generators for lhc physics", Physics Reports **504**, 145–233 (2011).

[9] G. Altarelli and G. Parisi, "Asymptotic freedom in parton language", Nuclear Physics B **126**, 298–318 (1977).

[10] T. N. Collaboration, R. D. Ball, V. Bertone, S. Carrazza, L. D. Debbio, S. Forte, P. Groth-Merrild, A. Guffanti, N. P. Hartland, Z. Kassabov, J. I. Latorre, E. R. Nocera, J. Rojo, L. Rottoli, E. Slade, and M. Ubiali, *Parton distributions from high-precision collider data*, 2017.

[11] A. Buckley, J. Ferrando, S. Lloyd, K. Nordström, B. Page, M. Rüfenacht, M. Schönherr, and G. Watt, "Lhapdf6: parton density access in the lhc precision era", The European Physical Journal C **75**, 10.1140/epjc/s10052-015-3318-8 (2015).

[12] S. Behnel, R. Bradshaw, C. Citro, L. Dalcin, D. S. Seljebotn, and K. Smith, "Cython: the best of both worlds", Computing in Science & Engineering **13**, 31–39 (2011).

[13]S. K. Lam, A. Pitrou, and S. Seibert, "Numba: a llvm-based python jit compiler", in Proceedings of the second workshop on the llvm compiler infrastructure in hpc, LLVM 15 (2015).

[14]J. C. Collins and D. E. Soper, "Angular distribution of dileptons in high-energy hadron collisions", Phys. Rev. D **16**, 2219–2225 (1977).

[15]S. Schumann and F. Krauss, "A parton shower algorithm based on catani-seymour dipole factorisation", Journal of High Energy Physics **2008**, 038 (2008).

[16]S. Hoeche, S. Schumann, and F. Siegert, "Hard photon production and matrix-element parton-shower merging", Physical Review D **81**, 10.1103/PhysRevD.81.034026 (2009).

[17]J.-C. Winter, F. Krauss, and G. Soff, "A Modified cluster hadronization model", Eur. Phys. J. C **36**, 381–395 (2004).

[18]E. Bothmann et al. (Sherpa), "Event Generation with Sherpa 2.2", SciPost Phys. **7**, 034 (2019).

[19]F. C. Porter, *Testing consistency of two histograms*, 2008.

Erklärung

Hiermit erkläre ich, dass ich diese Arbeit im Rahmen der Betreuung am Institut für Kern- und Teilchenphysik ohne unzulässige Hilfe Dritter verfasst und alle Quellen als solche gekennzeichnet habe.

Valentin Boettcher

Dresden, Juni 2020

Acknowledgements

I want to thank Frank Siegert, my thesis advisor, for his patience with which he endured my rapid fire of Skype messages and all the helpful explanations.

I also want to thank Christian Wiel for proof reading my thesis and giving me insightful hints. I am indepted to Falko Trojahn, who also has proof-read with the perspective of an “outsider”.